Section 10. Management and Control Subsystem

10.1 Introduction

The MACS is responsible for the system-level control and monitoring of LPS devices and processes, and provides the interface between the LPS and the operator. It produces LPS metadata files; maintains the metadata accounting tables; and generates the LPS Q&A, data received summary, and file transfer summary reports. The MACS initiates data capture automatically according to the contact schedule, initiates Level 0R processing, deletes or retains the LPS files, and sends DANs to EDC DAAC. It also is responsible for inserting or updating the LPS processing parameters and thresholds, contact schedules, and IAS parameters in the LPS database.

10.2 Design Overview

This section provides an overview of the MACS software design. The relationship between the MACS and other Landsat 7 subsystems is presented, along with a discussion of the assumptions, constraints, and considerations used in the design process.

10.2.1 Subsystem Software Overview

As shown in the MACS context diagram (Figure 10–1), the MACS interfaces with the RDCS, RDPS, MFPS, PCDS, IDPS, and LDTS. The roles of the MACS functions are as follows:

- The setup function provides the LPS operator with the capability of updating the LPS string configuration, contact schedules, data thresholds and parameters, and IAS parameters.

- The control function allows the operator to start/stop the manual raw data capture, start/stop copying data to a removable media, start/stop restaging data, and start/stop Level 0R data processing.

- The scheduled data capture function controls automatic data capture driven by the contact schedule from the Contact_Schedules table. It starts data capture when the current time from the operating system (minus a period of time sufficient for RDCS capture set up) is equal to a scheduled contact start time.

- The monitoring function receives device and processing status information, displays any processing anomalies, and alerts the operator when necessary. It provides the operator with the capability of viewing operational messages of selected severity level and viewing the LPS Journal file (a log file) at any time.

- The Metadata generation function creates a metadata file per subinterval after Level 0R data processing is completed. Q&A information is retrieved from the accounting tables of the Level 0R processing subsystems (i.e., RDPS, MFPS, PCDS, and IDPS) to produce metadata files.

- The reports function allows the operator to generate the LPS Q&A, data received summary, and file transfer summary reports.

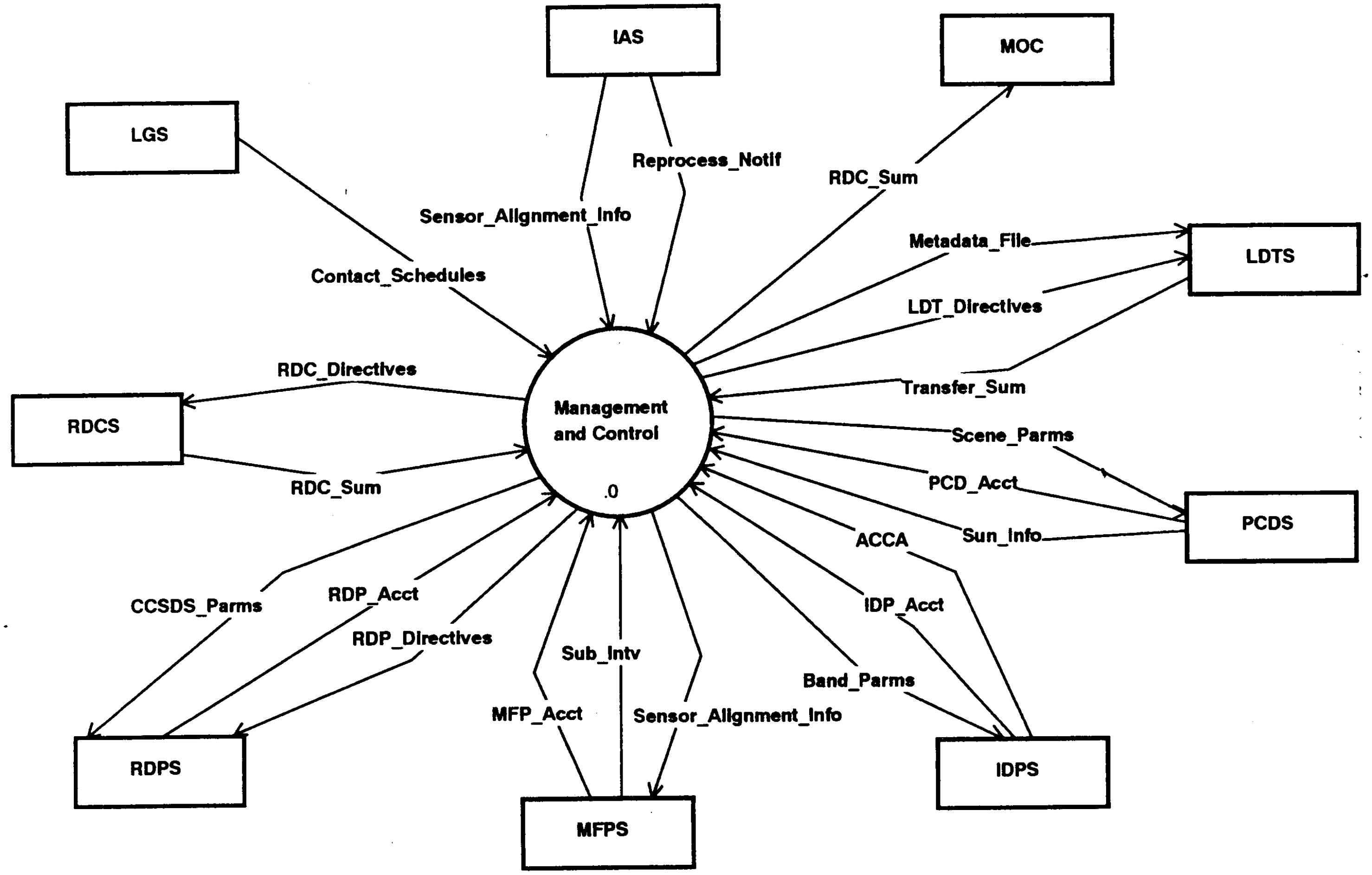


Figure 10–1. Management and Control Context Diagram

- The file management function allows the operator to enable/disable DAN transfer, resend DAN, retain/delete output files, or start/stop the DDN server.

- The test function allows the operator to write a test data file to the capture device.

- The shutdown function allows the operator to shutdown the LPS.

10.2.2 Design Considerations

This section presents the design drivers relevant to MACS software design. The assumptions, software reuse strategy, and required operational support that influenced the design of the MACS software are described.

10.2.2.1 Assumptions and Open Issues

The following assumptions have been made:

- The operator will not change the LPS configuration, LPS subsystem thresholds, or LPS subsystem parameters while data capture or Level 0R processing is running.

- The process identifier of the automatic data capture, Level 0R processing, and DDN server will be stored in the Process_Id table. This table will have the process task name and corresponding unique process identifier.

- At the time of LPS startup, the MACS will start the automatic data capture and DDN server. It will terminate the automatic data capture and DDN server when it shuts down the LPS.

- The LPS does not allow the operator to shut down the LPS once data capture or Level 0R processing is underway.

No issues remain open.

10.2.2.2 Operational Support

10.2.2.2.1 Start Up the MACS Software

The MACS will be started by an operator from a command line. During initialization, the MACS will

- Bring up the LPS GUI-based user interface, which lets the operator select different options from the menu.

- Bring up the main function for automatic, scheduled raw data capture.

- Start the database initialization task.

- Start the DDN server task that establishes server session for receiving DDNs, process DDNs, and send DDAs.

10.2.2.2.2 Avoid Abnormal Termination

If the database is unavailable at the time of MACS startup, the operator will not be able to bring up the user interface and will have to delete extraneous LPS startup processes from the command line.

Signal traps exist in every subsystem. They provide a means to gracefully terminate when fatal processing errors occur.

10.2.2.2.3 Generate Metadata file on a Subinterval basis

The MACS generates a metadata file for each subinterval in the contact period. Once all associated metadata files are generated, a request to send a DAN is sent to the LDTS.

10.2.2.2.4 Start Data Capture

The MACS has an automatic data capture function that is driven by the contact schedule in the Contact_Schedules table. This function starts the RDCS when the current time from the operating system is equal to the scheduled start time (minus a period of time sufficient for RDCS capture setup).

10.2.2.2.5 LPS System Monitoring

The MACS provides the operator with tools to monitor the LPS operations for system failure. It also receives the processing statuses from the subsystems and displays them on the operator's screen. In addition, the operator may view the LPS Journal file for a complete picture of any Level 0R process.

10.2.2.3 Software Reuse Strategy

This section identifies the external components that may be reused by the MACS, as well as common components of the MACS software that may be useful to other Landsat 7 subsystems. Tables 10–1 through 10–3 list the component type and description for each reusable component.

10.2.3 Subsystem Error Handling

This section addresses the errors and processing exceptions that may be encountered by the MACS. The severity classification of each error and the system responses to each of the errors are described in Tables 10–4 and 10–5, respectively.

Table 10–1. Component Type

Component Type	Description
Design	The algorithm only may be reused.
Design/code	The algorithm and the code may be reused.
New development	The unit has not yet been developed.

Table 10–2. Ease of Use

Ease of Use	Description
Major modifications required	Substantial modification is required to reuse the component.
Minor modifications required	Minor modification is required to reuse the component.
No modifications	No modification is required to reuse the component.

Table 10–3. Reusable Components

Reusable Component	Type	Ease of Use
Syslog function File used to log all the LPS messages	Design/code	Minor modifications required

Table 10–4. Severity Classifications

Severity Classification	Description
REQ REENTRY	Do not allow the error to enter the system. Require reentry of the data.
NOTIFY	Notify operations personnel of unexpected input. This is not reported as a definite error, merely a potential error. Continue processing as normal.
WARNING	Warn operations personnel of erroneous input, but continue processing as normal.
AL/CON	Generate an alarm, but continue processing as normal for all data streams.
AL/HALT/PROC	Generate an alarm and halt processing. Requires operator intervention to reprocess entire contact period.

Table 10–5. System Responses

Error Description	Severity	System Response
Failure to connect to database	AL/HALT/PROC	If the MACS is unable to connect to the database, the subsystem will halt processing.
Insert or update the contact schedules	REQ REENTRY	When the operator enters the contact schedule, each field is validated and requires reentry if it is invalid or out of range.
Insert or update the subsystem thresholds tables	REQ REENTRY	When the operator enters the thresholds for a subsystem, each field is validated and requires reentry if it is out of range.
Insert or update the subsystem parameters tables	REQ REENTRY	When the operator enters the parameters for a subsystem, each field is validated and requires reentry if it is out of range.

10.3 Subsystem Design

10.3.1 Top-Level Model

The top-level model of the MACS is shown in Figure 10–2. mac_ui_MainMenu is the driver of the MACS software, which is invoked by the operator from the command line. When mac_ui_MainMenu starts up, the main menu of LPS is displayed on the screen. The menu has seven options: Setup, Test, Control, Monitor, FileMgmt, Reports, and Shutdown. Each of these menu items have submenus. Each menu option on the menu, when chosen, invokes a user interface function as follows:

·	Setup	mac_ui_MainSetup
·	Test	mac_ui_MainTest
·	Control	mac_ui_MainControl
·	Monitor	mac_ui_MainMonitor
·	FileMgmt	mac_ui_MainFileMgt
·	Reports	mac_ui_MainReports
·	Shutdown	mac_ui_MainShutdown

10.3.2 Detailed Module Design

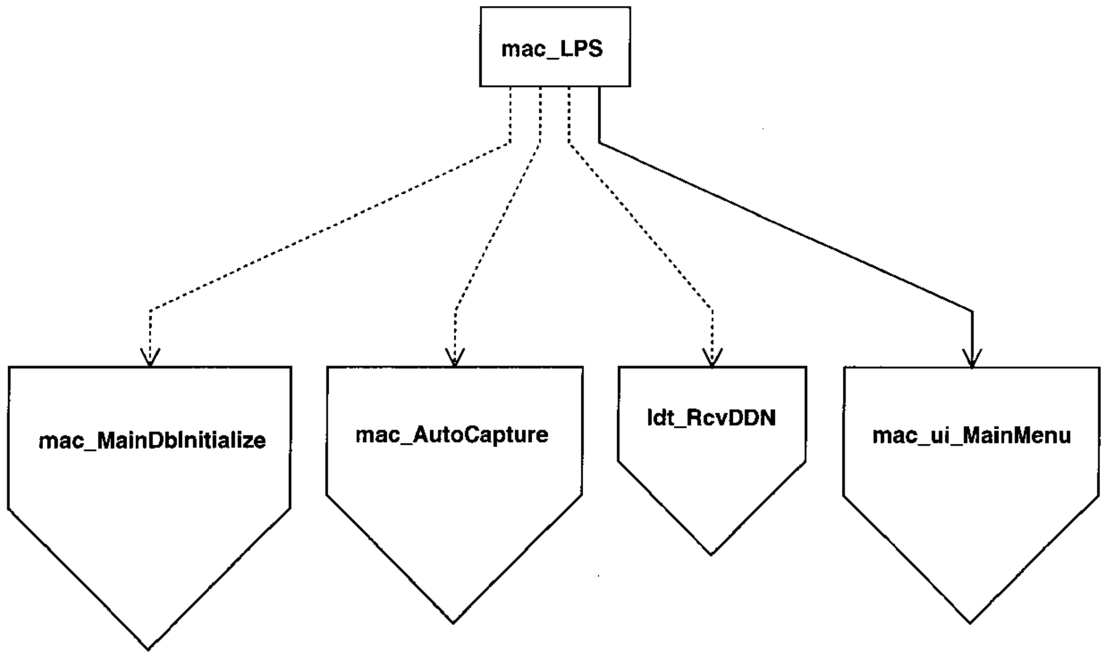
10.3.2.1 mac_ui_MainMenu

mac_ui_MainMenu (Figure 10–3) represents the MACS software for LPS management and Level 0R processing. It displays the main user interface menu on the screen. This main menu has seven different options, as described in the following subsections.

10.3.2.1.1 mac_ui_MainSetup

mac_ui_MainSetup (Figure 10–4) represents the Setup menu option of the main menu. From this menu the operator may select from the following options:

- View/Edit Capture Source – allows operator to change capture source identifier
- Ingest Contact Schedule – allows operator to select a contact schedule file and which string(s) the contact schedule will be read into
- View/Edit Contact Schedule – allows operator to modify the contact schedule information
- Load IAS Parameter File – allows operator to select an IAS parameter file
- Propagate L0R Parameters – allows operator to select string(s) for propagation of L0R parameters
- View/Edit L0R Parameters – allows operator to modify L0R parameters
- View/Edit L0R Thresholds – allows operator to modify L0R thresholds



Figurc 10–2. MACS Top-Lcvcl Structurc Chart

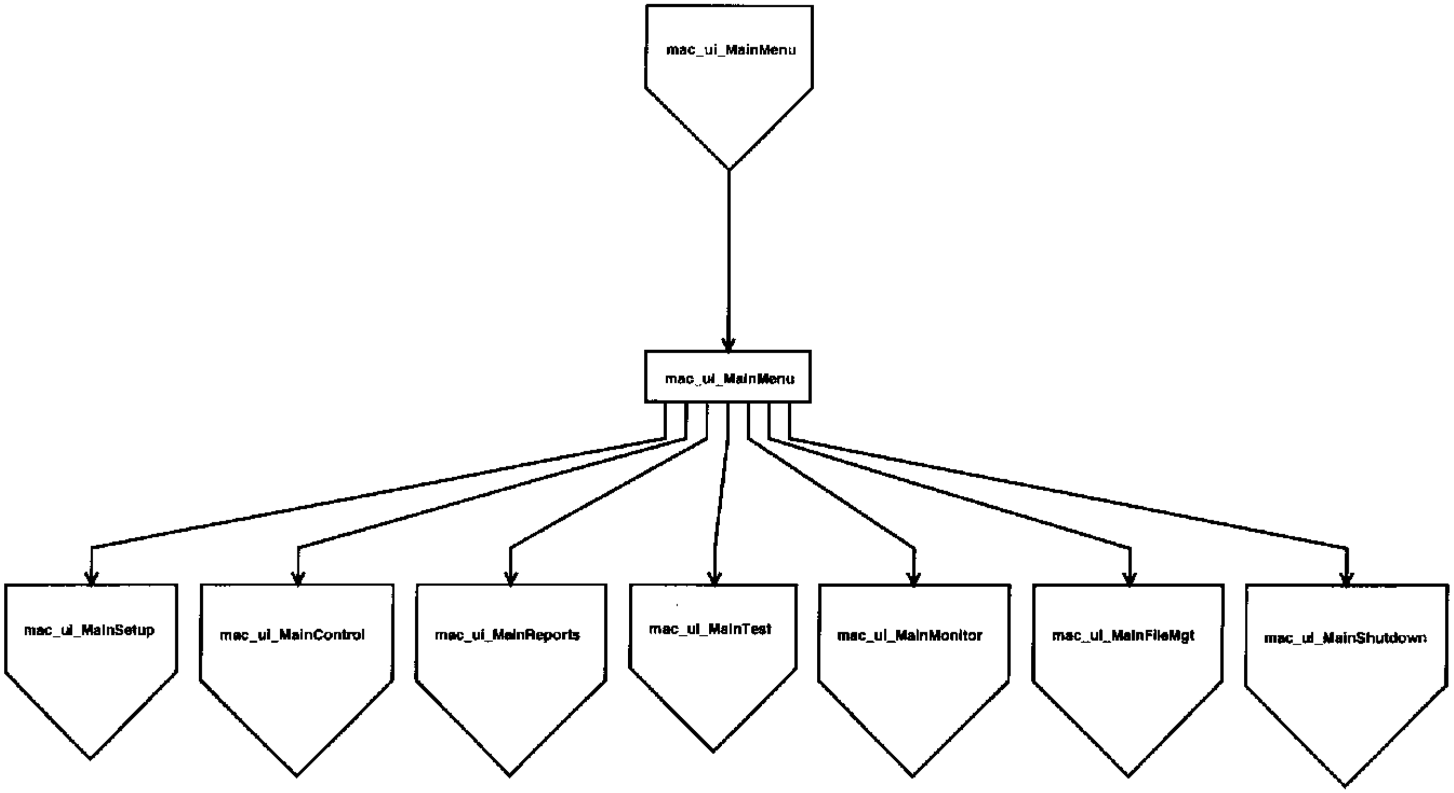


Figure 10–3. *mac_ui_MainMenu* Structure Chart

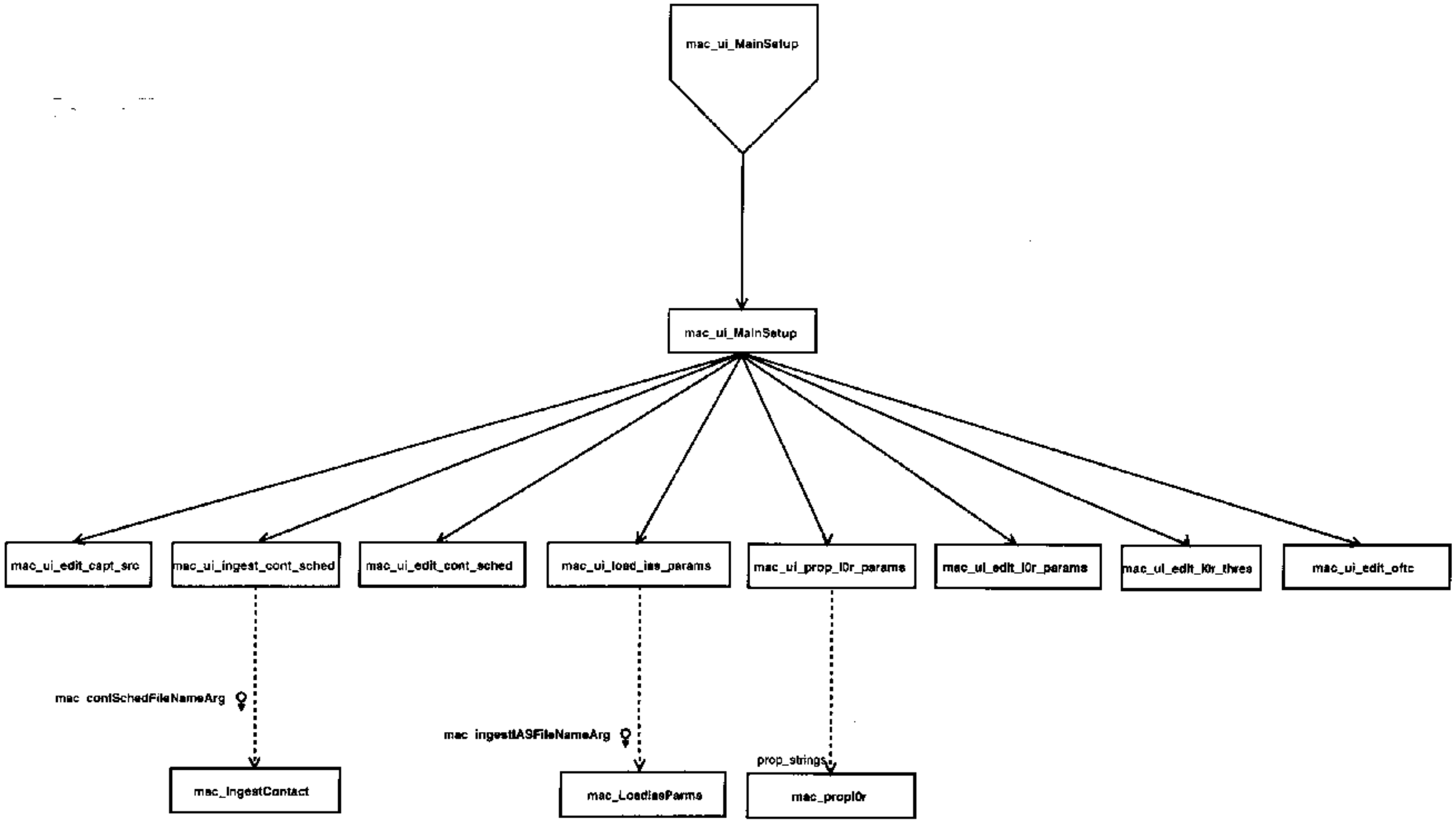


Figure 10–4. mac_ui_MainSetup Structure Chart

- View/Edit Output File Transfer Config – allows operator to modify the output file transfer configuration

10.3.2.1.2 mac_ui_MainControl

mac_ui_MainControl (Figure 10–5) represents the Control menu on the main menu. It provides the operator with different options, such as

- Start Capture
- Stop Capture
- Start L0R Processing
- Stop L0R Processing
- Start/Stop Copy to Tape
- Start/Stop Restage
- Start/Stop Auto Capture
- Generate Tape Label

Start data capture is the manual option to start the raw data capture, and it forks rdc_Main. Stop data capture is a manual option to stop data capture before the scheduled stop time. Start restage provides the capability to copy the files for a requested contact period from the tape to the disk for reprocessing. Stop restage stops the data copy process for the requested contact period. Start tape copy provides the capability to copy the disk files for the requested contact period to the tape. Stop tape copy stops the tape copy in progress for the requested contact period. Start Level 0R processing forks rdp_Main, mfp_Main, pcd_Main, and idp_Main; generates metadata; and calls ldt_SendDAN.

Once the processing is done for a contact period, metadata files are generated for each subinterval and ldt_SendDAN is called to generate and send a DAN to EDC DAAC. Stop Level 0R processing is the manual option to stop the processing before it is completed. Stop Level 0R processing invokes mac_MainShutdownL0R, which calls mac_MainAbortL0R, mac_MainRollbackL0R, and mac_MainCleanupL0R. mac_MainAbortL0R determines which child processes are active and terminates them. mac_MainRollbackL0R rolls the database back to the state that existed prior to activation of the current Level 0R processing. Once the database is rolled back, mac_MainCleanupL0R is called to deallocate the shared memory and delete all files created during this Level 0R processing. Start auto data capture invokes mac_AutoCapture, which activates the automatic raw data capture function. Generate tape label invokes rdc_GenLabel.

10.3.2.1.3 mac_ui_MainReports

mac_ui_MainReports (Figure 10–6) represents the Reports option on the main menu. The data Q&A reports available under this option include data received summary, LPS Q&A, and file transfer summary. Data Received Summary lets the operator display or print the summary report

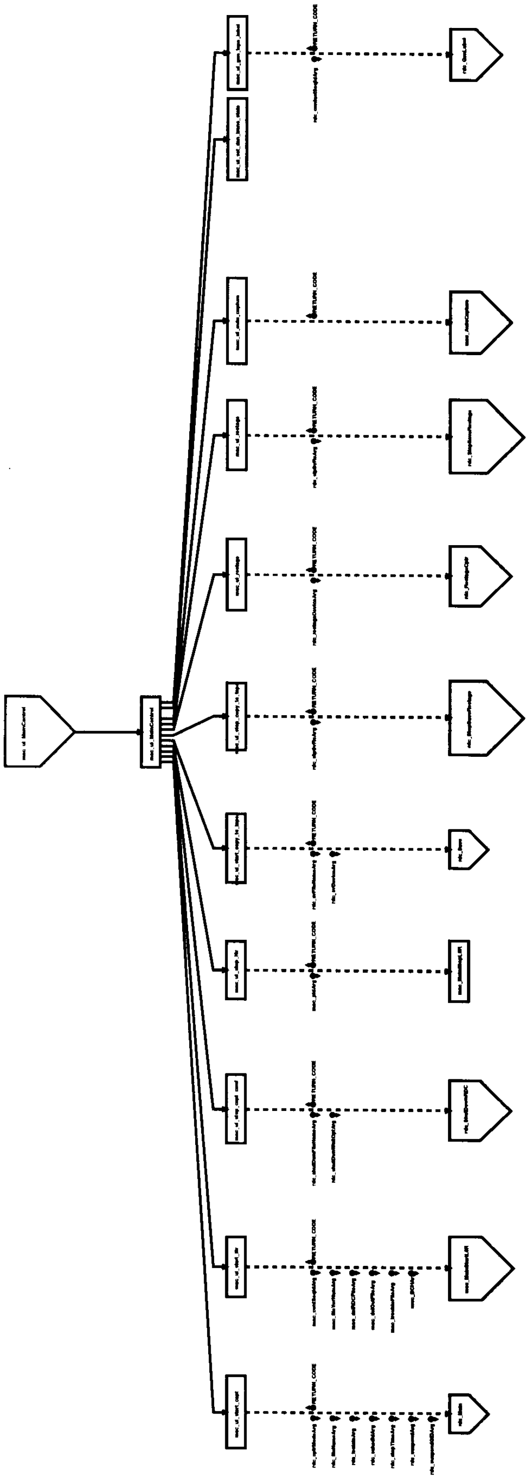


Figure 10–5. mac_ui_MainControl Structure Chart

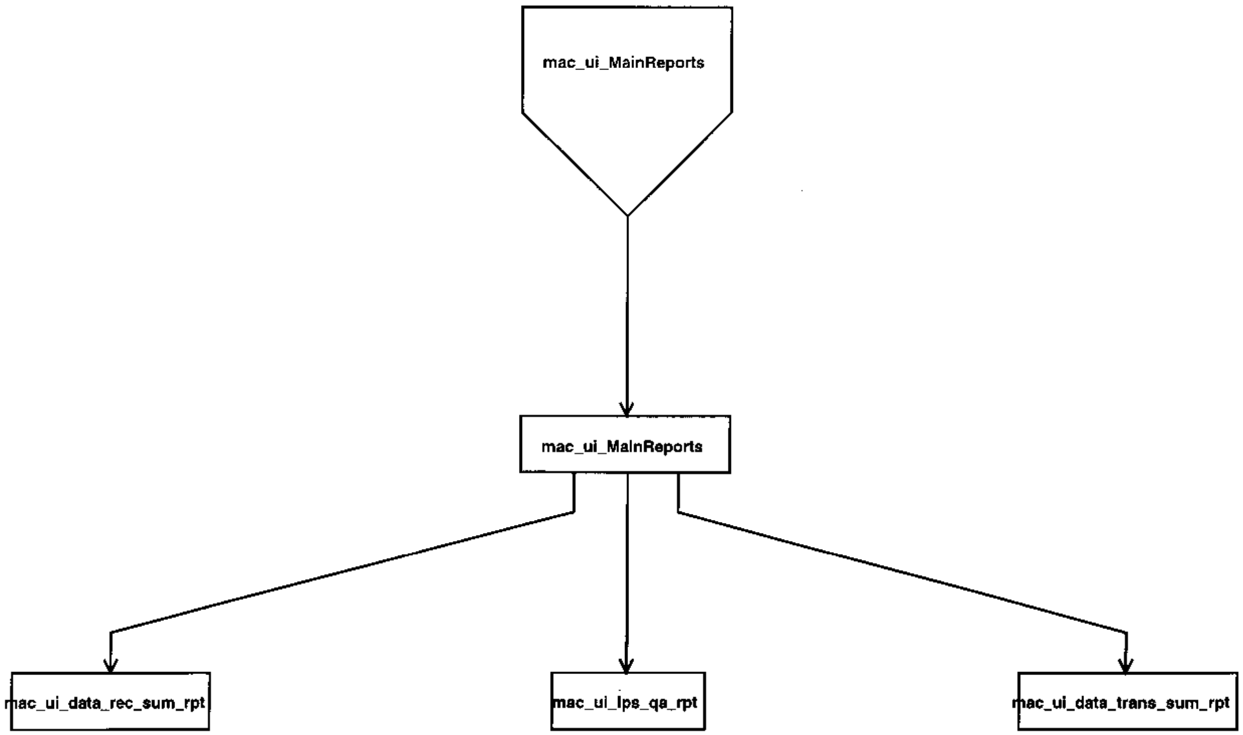


Figure 10–6. mac_ui_MainReports Structure Chart

for one contact period. LPS Q&A lets the operator display or print the raw wideband data Q&A report based on a contact period and its subintervals. File Transfer Summary lets the operator display or print the LPS file transfer summary report based on the requested begin and end time.

10.3.2.1.4 mac_ui_MainMonitor

LPS operations can be monitored by selecting the Monitor option that invokes mac_ui_MainMonitor (Figure 10–7). The operator can either view the LPS Journal file or monitor the ongoing operations displayed on the screen.

10.3.2.1.5 mac_ui_MainFileMgt

mac_ui_MainFileMgt (Figure 10–8) has the following options available:

- DAN Transfer State – allows operator to enable or disable the DAN transfer state
- Resend Suspended DAN – sets DAN transfer state to enable and invokes resendsuspdan
- Resend Failed DAN – invokes senddan for the operator-selected contact period
- Output File Set Management – allows operator to retain or delete output files for the selected contact period
- Delete Raw File – allows operator to select a raw data file to be deleted and invokes rdc_DeleteFiles
- Start/Stop DDN Server – invokes rcvddn to start the DDN server or stopddn to stop the DDN server

10.3.2.1.6 mac_ui_MainTest

mac_ui_MainTest (Figure 10–9) has a submenu item, Send Data, that lets the operator write a test data file to the capture device.

10.3.2.1.7 mac_ui_MainShutdown

The operator exits the user interface by selecting the Shutdown option that invokes mac_ui_MainShutdown (Figure 10–10).

10.3.2.2 mac_MetaDataGen

mac_MetaDataGen (Figure 10–11) is responsible for generating metadata files for each subinterval that makes up the specified contact period. This module is called from within the start Level 0R process after accounting information from all subintervals is available for the specified contact period. A metadata file contains a file header, subinterval information, and multiple scene accounting data belonging to a subinterval.

When mac_MetaDataGen is activated, a metadata file is created for each subinterval, as indicated by the Sub_Intv table. Each metadata file is generated using information extracted from the LPS Sub_Intv, Bands_Present, RDC_Acct, MFP_Acct, PCD_Acct, PCD_Scene_Acct, IDP_Acct, LPS_File_Info, MFP_MJF_Acct, PCD_MJF_Acct, Band_Gain_States, and LPS_Configuration database tables. Separate functions within this module format the extracted data and place it within the header and data portions of the generated metadata file.

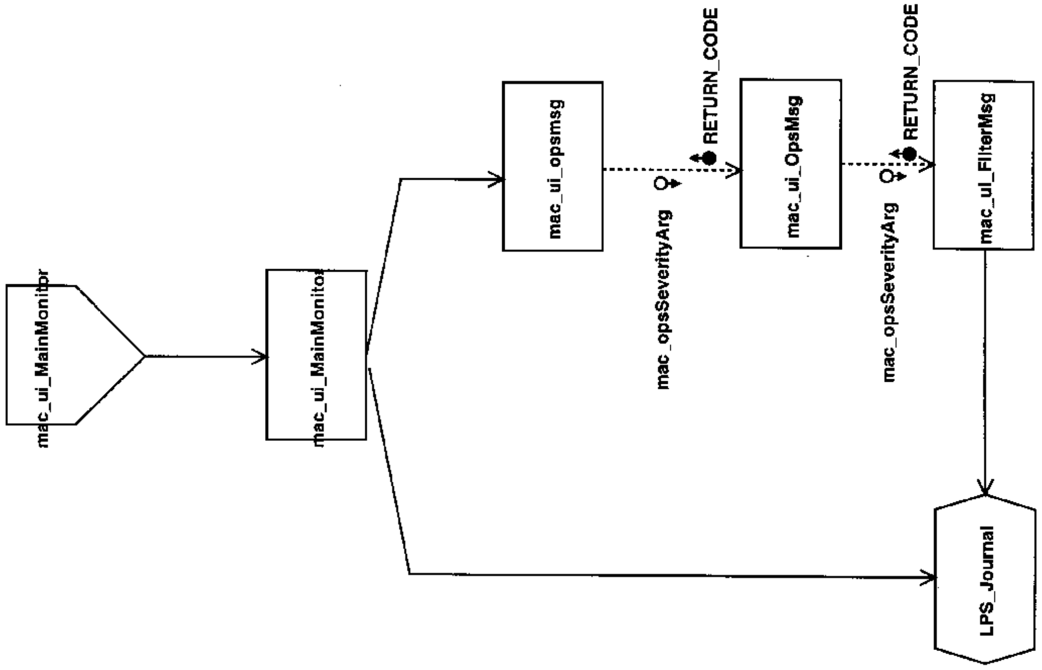


Figure 10–7. mac_ui_MainMonitor Structure Chart

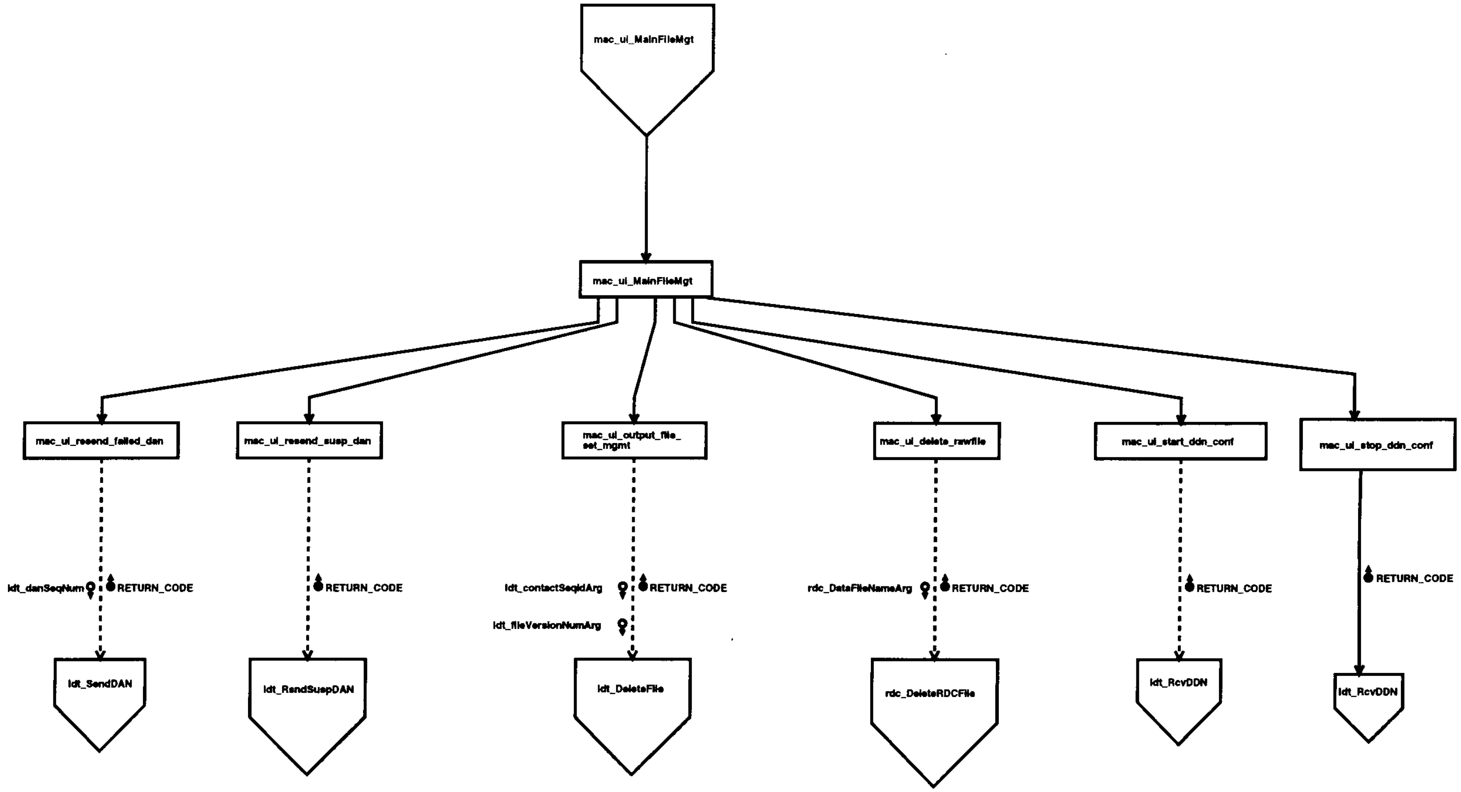


Figure 10–8. mac_ui_MainFileMgt Structure Chart

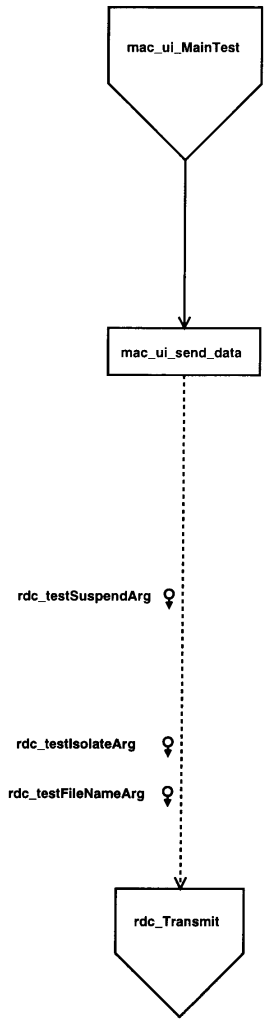


Figure 10–9. mac_ui_MainTest Structure Chart

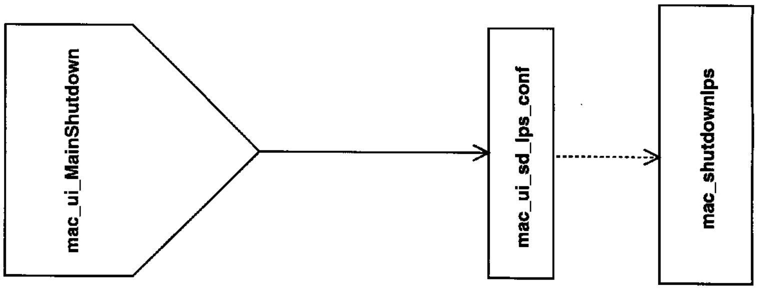
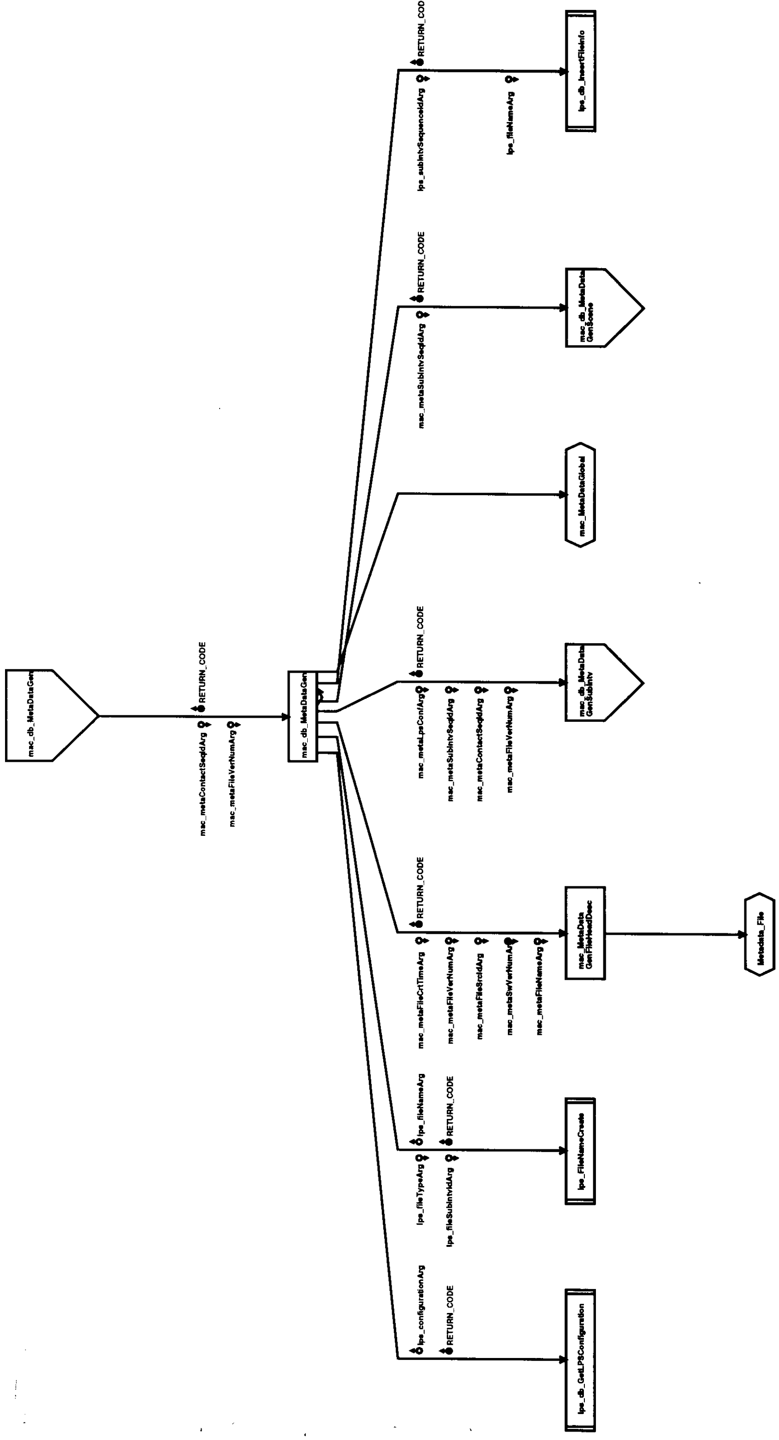


Figure 10–10. mac_ui_MainShutdown Structure Chart

Figure 10–11. mac_MetaDataGen Structure Chart



On completion of metadata file generation for the indicated contact period, lps_db_InsertFileInfo is invoked to insert the metadata filename into the LPS_File_Info table.

10.3.2.3 mac_MetaDataGenFileHeadDesc

mac_MetaDataGenFileHeadDesc is responsible for writing the file header information into the metadata file. It retrieves information from the LPS_Configuration table.

10.3.2.4 mac_MetaDataGenSubIntv

mac_MetaDataGenSubIntv (Figure 10–12) extracts the subinterval information from the Sub_Intv, Bands_Present, MFP_Acct, PCD_Acct, RDC_Acct, IDP_Acct, and LPS_File_Info tables and writes it into the metadata file.

10.3.2.5 mac_MetaDataGenScene

mac_MetaDataGenScene (Figure 10–13) retrieves the scene information that makes up the subinterval from the PCD_Scene_Acct, IDP_Acct, and Valid_WRS_Parms tables and writes it into the metadata file.

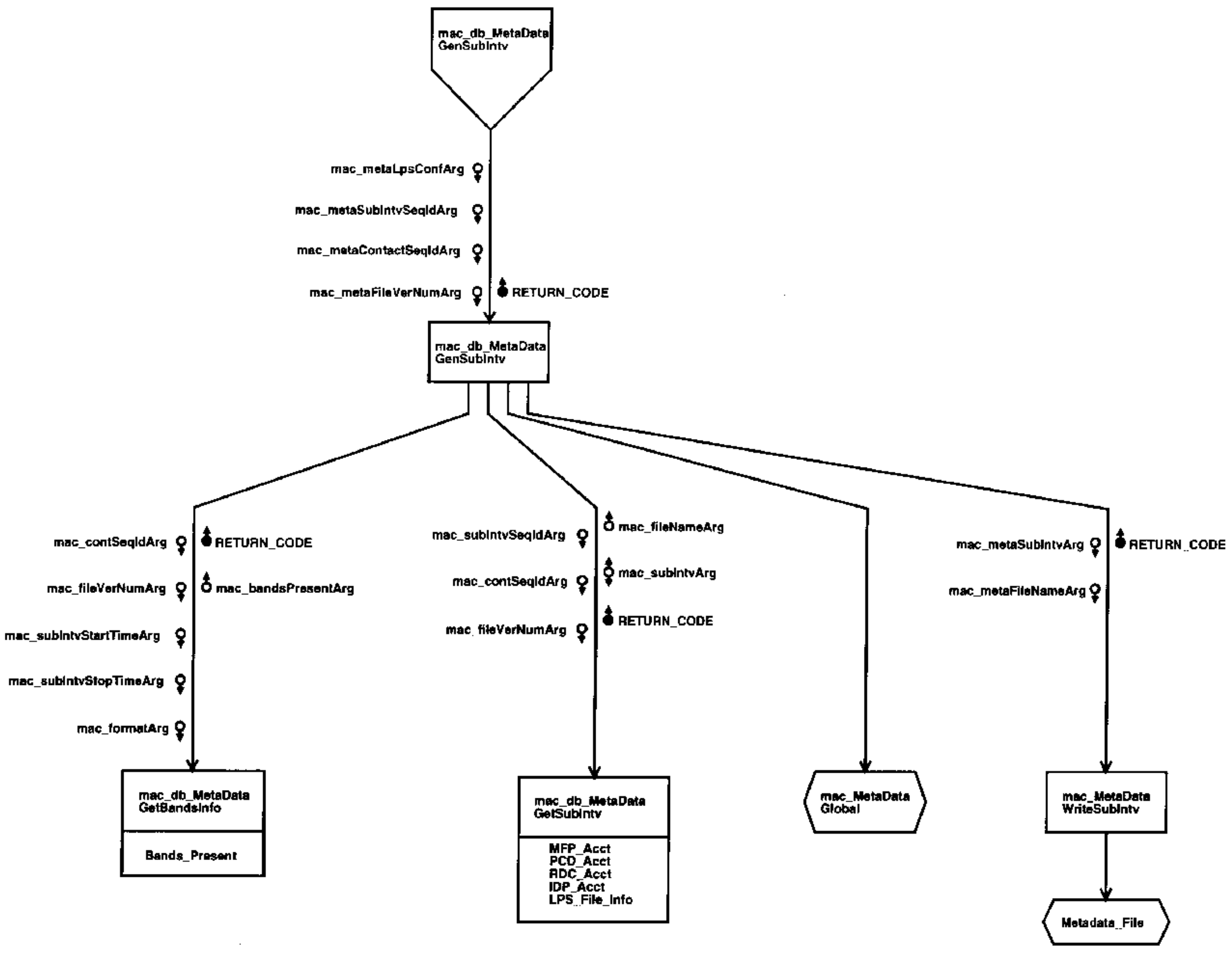


Figure 10–12. mac_MetaDataGenSubIntv Structure Chart

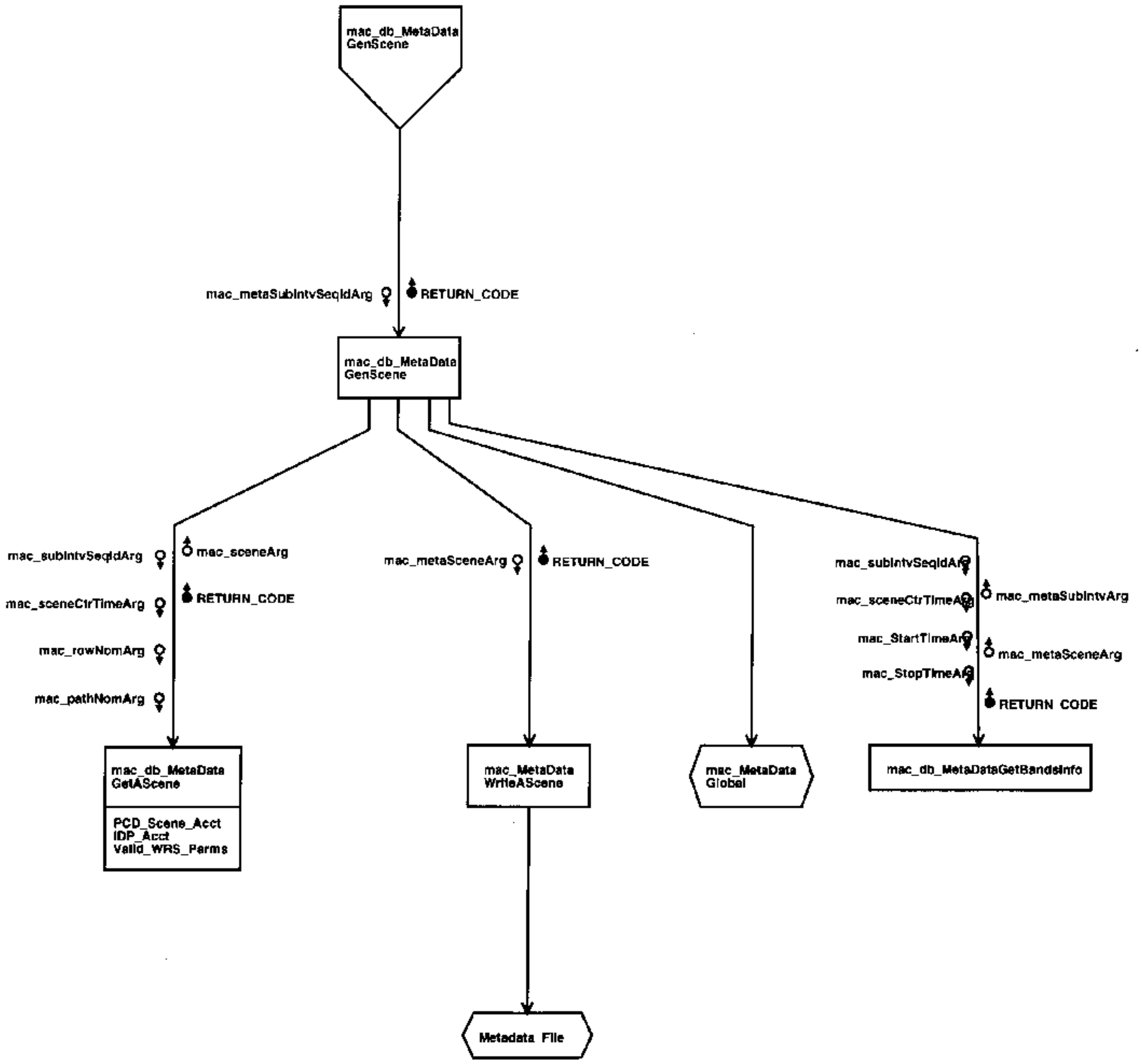


Figure 10–13. mac_MetaDataGenScene Structure Chart

Section 11. LPS Data Transfer Subsystem

11.1 Introduction

The LDTS is responsible for

- Informing EDC DAAC when sets of LPS output files become available[1] for transfer (sending DANs)
- Receiving from EDC DAAC the DAA confirming that EDC DAAC received the DAN and identifying any DAN errors
- Receiving from EDC DAAC indications (DDNs) that EDC DAAC could or could not successfully transfer available files
- Accumulating file transfer summary in the database
- Deleting LPS output files after they have been acknowledged in DDNs as having been successfully transferred or when delete requests are received from the MACS
- Marking file sets for retention
- Resending suspended DANs when DAN transfer is enabled or resending a failed DAN on request by the operator

11.2 Design Overview

This section provides an overview of the LDTS software design. The relationships between the LDTS and other Landsat 7 subsystems and EDC DAAC are presented, along with a discussion of the assumptions, constraints, and considerations used in the design process.

11.2.1 Subsystem Software Overview

As shown in the LDTS context diagram (Figure 11–1), the LDTS interfaces with the MACS to receive requests to send and resend DANs, receive notification of the enabling or disabling of DAN transfer, receive requests to delete or retain output files, and report receipt of DDNs from EDC DAAC.

The LDTS operates on LPS output file sets. A file set is the set of all output files produced when the raw data captured for one contact period is processed once. A file set is uniquely identified by a contact identifier and a file version identifier. Any subsequent processing of the same raw data will create new and distinct file sets. The contact identifier will be the same, but the file version identifier will be different. For example, if the raw data for a particular contact is processed and then reprocessed twice, three distinct output file sets will be produced. The LDTS will then perform its processing on these file sets independently of each other.

[1] NOTE: The LDTS simply informs EDC DAAC of file availability, it does not transfer output files.

A file set becomes known to the LDTS when the MACS first requests that the LDTS send a DAN for it. The LDTS records in the database the time that this file set became available, which

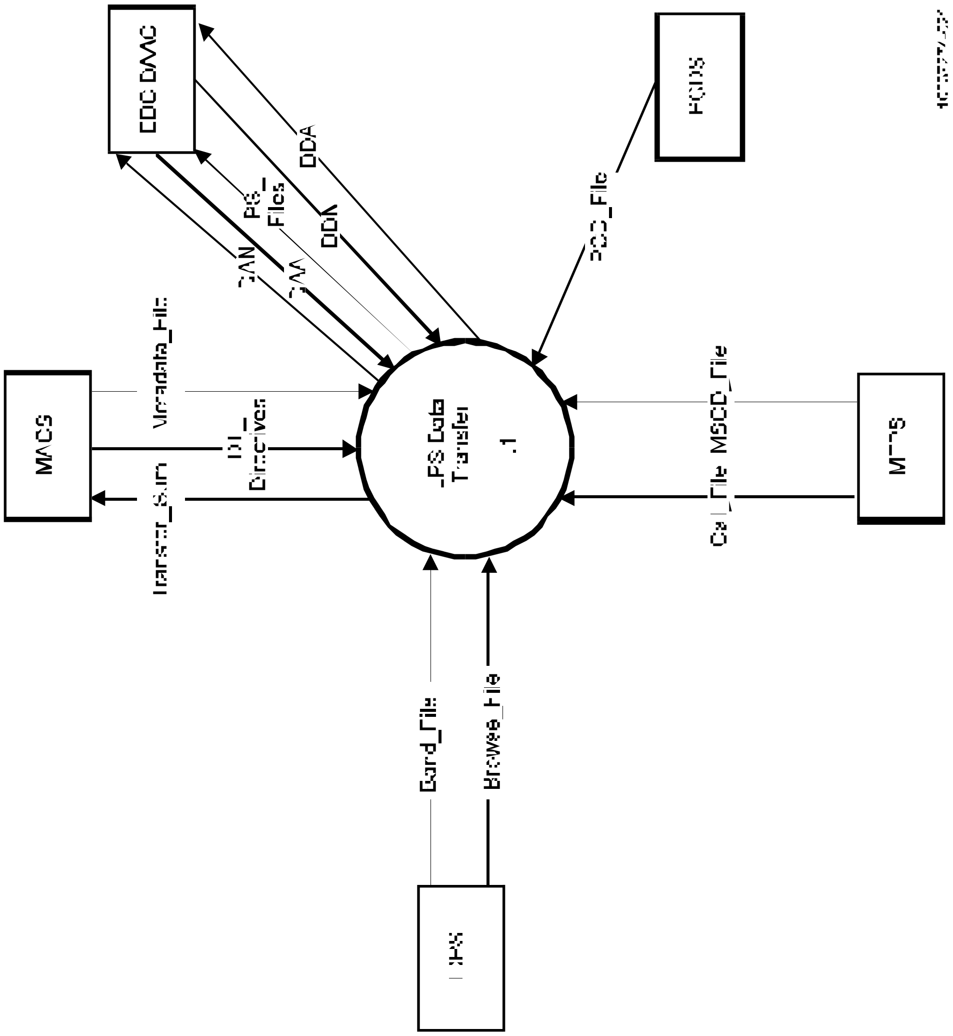


Figure 11–1. LPS Data Transfer Context Diagram

is important information in generating file transfer summary reports. The LDTS then creates a DAN for the file set and transmits it to EDC DAAC. A file set is made up of file groups. Associated with each contact identifier and file version identifier pair are a unique set of subinterval identifiers. Each subinterval identifier becomes a file group identifier in the LDTS. Each file group consists of the LPS output files associated with a particular file group (subinterval) identifier.

Once a DAN is sent to EDC DAAC, the LDTS will wait for the DAA from EDC DAAC to confirm EDC DAAC has received the DAN and to identify any DAN errors. DAN status information is recorded into the database for use in resending the DAN and generating the file transfer summary report.

In response to a DAN, EDC DAAC sends to the LDTS a DDN that lists the filenames in the file set indicated in the DAN and, for each file, an indication of whether or not EDC DAAC was able to transfer that file. Files in the DDN that have transfer errors associated with them are reported by the LDTS to the MACS. Communication-related errors and other errors also are reported, which gives the LPS operator the opportunity to correct the error(s) and the option to issue a resend request or a delete request to the LDTS. File groups containing files all successfully transferred (transferred file groups) are assigned a DDN receipt time in the database. File groups containing at least one file not successfully transferred are considered to be failed, and the operator must decide, based on the reason for failure, what action to take (delete file set or fix problem and resend DAN). DDN receipt times are used to generate the file transfer summary report. They also serve to indicate which file groups are transferred.

When DDNs are received, files in transferred file groups in file sets that have not been marked for retention will be deleted from storage. The MACS may, at any time, issue to the LDTS a request for a file set to be retained. When this occurs, the file set enters a state wherein its files may no longer be deleted except by a request from the MACS. The MACS may, at any time, send a request to the LDTS to delete any file set.

When a DAN for a file set is sent for the first time, it will include information on every file in every file group in the file set. However, if and when that DAN is resent to EDC DAAC, only information about the contents of failed groups in that file set are included in the DAN. Eventually, all file groups in a file set will have DDN receipt times. When this occurs, an indication that the entire file set has been successfully or unsuccessfully transferred is stored in the database. If the file set has been transferred successfully, it is no longer considered to be active in the LDTS.

Periodically, the LDTS checks the DAN acknowledgment time stamp and DDN receive time stamp to make sure that every DAN sent to EDC DAAC has corresponding DAA and DDN messages from EDC DAAC. If either one is missing after a timeout duration is expected, the LDTS will report the missing DAA or DDN to the MACS.

Before any DAN is transmitted, the LDTS determines whether or not DAN transfer is enabled. The DAN transfer state (enabled or disabled) is maintained by the MACS. DANs are not sent to EDC DAAC when DAN transfer is disabled. Instead, any request received by the LDTS to send or resend a DAN for a file set results in the LDTS marking that file set as having a suspended DAN.

An indication is returned to the MACS that the request resulted in a suspended DAN for the given file set. When the MACS enables DAN transfer, the LDTS searches the active file sets in the database for file sets that have suspended DANs. Each suspended DAN is sent or resent.[2]

11.2.2 Design Considerations

This section presents the design drivers relevant to the LDTS software design. The assumptions, software reuse strategy, and required operational support that influence the design of the LDTS software are described.

11.2.2.1 Assumptions and Open Issues

The LDTS subsystem design is based on the following assumptions:

- Before the MACS issues a request for the LDTS to send or resend a DAN, the LPS File Information database table will already contain information about the files associated with the subinterval identifiers that are associated with a file set identified in the send/resend request.
- A socket interface will be used to send DANs to and receive DDNs from EDC DAAC.
- A new DAN will always be generated for resending a failed DAN.
- Short DDAs will be sent in response to both long and short DDNs as requested by EDC DAAC development personnel.
- The LDTS DDN server will only handle one connection from EDC DAAC at any given time.
- The LDTS can have multiple connections to EDC DAAC for sending multiple DANs concurrently.

No open issues remain unresolved for the LDTS.

11.2.2.2 Operational Support

This section presents components of the software design that support the normal and contingency operations of the LDTS.

11.2.2.2.1 Start Up the LDTS Software

The LDTS is composed of six independent main programs:

1. ldt_SendDAN sends and resends DANs to EDC DAAC
2. ldt_DeleteFiles handles requests from the MACS and the LDTS DDN server to delete file sets
3. ldt_RetainFiles handles requests from the MACS to retain file sets

[2] A DAN is considered to be sent if, at the time of the request, no LDTS state information is provided on the given file set. If such state information does exist, the DAN will be resent.

4. ldt_RcvDDN, the LDTS DDN server, watches for incoming DDNs and parses and processes them when they arrive from EDC DAAC

5. ldt_RsndSuspDANs is triggered by the MACS enabling DAN transfer and finds all file sets for which DANs have been suspended, updates them to indicate that they no longer have suspended DANs, and calls ldt_SendDAN to resend each of the associated DANs

6. ldt_StopDDN terminates the LDTS DDN server on request from the MACS

11.2.2.2.1.1 ldt_SendDAN

ldt_SendDAN will be "forked" as a child process by the MACS whenever a DAN is sent or resent. ldt_SendDAN will

- Connect to the database and remain connected throughout its execution
- Retrieve the EDC DAAC Internet Protocol (IP) hostname and port number from the database to establish a communication link to EDC DAAC
- Retrieve the contact period Level 0R output filenames to format a DAN and send it to EDC DAAC
- Wait for the DAA message from EDC DAAC and record the DAN status into the database

If unable to obtain any of the services requested, create a DAN, or send the DAN to EDC DAAC, ldt_SendDAN will discontinue processing and will log the condition in the LPS Journal and indicate to the MACS the error condition in its return value.

11.2.2.2.1.2 ldt_DeleteFiles

ldt_DeleteFiles will be "forked" as a child process by the MACS or ldt_RcvDDN. ldt_DeleteFiles will

- Connect to the database and remain connected throughout its execution
- Delete the contact period file set or file group(s) automatically after EDC DAAC confirms the file set or file group(s) were successfully transferred and archived
- Manually delete a contact period output file set or an output file group, or individual output files
- Record the file deletion status of an output file set, or an output file group, or individual output files in the database

If unable to connect to the database or delete files, ldt_DeleteFiles will log the error in the LPS Journal, discontinue processing, and indicate to the MACS the error condition in its return value.

11.2.2.2.1.3 ldt_RetainFiles

ldt_RetainFiles will be implemented by the MACS during user interface. ldt_RetainFiles will mark or unmark the contact period file set for retention. If marked for retention, the file set will not be deleted automatically.

11.2.2.2.1.4 ldt_RcvDDN

ldt_RcvDDN will be “forked” as a child process by the MACS when LPS is started. ldt_RcvDDN will

- Connect to the database and remain connected throughout its execution
- Accept one and only one connection from EDC DAAC for receiving the DDN message
- Process the DDN and verify the status of data transfer and archival at EDC DAAC
- Record the data transfer status for a contact period file set, file groups, and individual output files into the database
- Periodically report the health message to the MACS

If unable to receive or process a DDN, ldt_RcvDDN will log the error in the LPS Journal and continue receiving and processing the next DDN.

11.2.2.2.1.5 ldt_RsndSuspDANs

ldt_RsndSuspDANs will be “forked” as a child process by the MACS. ldt_RsndSuspDANs will

- Connect to the database and remain connected throughout its execution
- Retrieve all suspended DANs from the database and resend them to EDC DAAC

If unable to connect to the database or resend the suspended DANs, ldt_RsndSuspDANs will log the error in the LPS Journal, discontinue processing, and indicate to the MACS the error condition in its return value.

11.2.2.2.1.6 ldt_StopDAN

ldt_StopDDN will be invoked by the MACS to terminate the LPS DDN server. ldt_StopDDN will

- Connect to the database and remain connected throughout its execution
- Verify if a DDN is actively being processed; if it is, ldt_StopDDN will wait until the DDN processing is complete and then terminate the DDN server

If unable to connect to the database, ldt_StopDAN will log the error in the LPS journal and terminate the DDN server any way.

11.2.2.2.2 Avoid Abnormal Termination

Four LDTS main programs (ldt_SendDAN, ldt_DeleteFiles, ldt_RcvDDN, and ldt_RsndSuspDANs) each set a signal trap to catch any signals that may cause abnormal termination. If a signal is trapped, a signal handler is executed to clean up the temporary files and disconnect from the database.

11.2.2.2.3 Support Reprocessing of Data

The LDTS processes file sets representing raw data processed for the first time in exactly the same manner as file sets representing reprocessed raw data.

11.2.2.3 Software Reuse Strategy

This section identifies external components that may be reused by the LDTS, as well as common components of LDTS software that may be useful to other Landsat 7 subsystems. Tables 11–1 through 11–3 list the component type and the ease of use classification for each reusable component.

Table 11–1. Component Type

Component Type	Description
Design	The algorithm only may be reused.
Design/code	The algorithm and the code may be reused.
New development	The unit has not yet been developed.

Table 11–2. Ease of Use

Ease of Use	Description
Major modifications required	Substantial modification is required to reuse the component.
Minor modifications required	Minor modification is required to reuse the component.
No modifications	No modification is required to reuse the component.

Table 11–3. Reusable Components

Reusable Component	Type	Ease of Use
Data Distribution Facility (DDF) consumer simulator code DDF simulator code	Design/code	Some units require minor modifications and others require major modifications

11.2.3 Subsystem Error Handling

This section addresses the errors and processing exceptions that may be encountered by the LDTS. Tables 11–4 and 11–5 describe the severity classification of each error and the system response to each of the errors, respectively.

Table 11–4. Severity Classifications (1 of 2)

Severity Classification	Description
REQ REENTRY	Do not allow the error to enter the system. Require reentry of the data.

NOTIFY	Notify operations personnel of unexpected input. This is not reported as a definite error, merely a potential error. Continue processing as normal.
WARNING	Warn operations personnel of erroneous input, but continue processing as normal.

Table 11–4. Severity Classifications (2 of 2)

Severity Classification	Description
AL/CONT	Generate an alarm, but continue processing as normal for all data streams.
AL/HALT	Generate an alarm and halt processing.
AL/HALT/PROC	Generate an alarm and halt processing. Requires operator intervention to reprocess entire contact period.

Table 11–5. System Responses

Error Description	Severity	System Response
Failure to set up signal catcher in any LDTS main program	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
Failure to connect to database by any LDTS main program	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
Failure of ldt_SendDAN or ldt_RcvDDN to obtain valid parameters from database	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
Failure of any LDTS routine to disconnect from database	WARNING	Log error in LPS Journal.
Failed attempt to read from database	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
Failed attempt to write to database	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
Failure to establish communications channel(s) with EDC DAAC	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
Repeated failures reading from communications channel(s) with EDC DAAC	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
Repeated failures writing to communications channel(s) with EDC DAAC	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
DDN received indicating transfer problems with one or more files	AL/CONT	Log error in LPS Journal.
DAN times out	AL/CONT	Log error in LPS Journal.
DDN received does not list all files that were in DAN	AL/CONT	Log error in LPS Journal.
Files indicated in delete request do not exist on LPS host machine	WARNING	Log error in LPS Journal.

Data capture is running	AL/HALT	Log error in LPS Journal, discontinue processing, return error code.
DAN send/resend request received when DAN transfer is disabled	NOTIFY	Log error in LPS Journal.

11.3 Subsystem Design

11.3.1 Top-Level Model

The top-level model of the LDTS is shown in Figure 11–2. The LDTS is composed of six independent main programs:

1. ldt_SendDAN sends and resends DANs to EDC DAAC.
2. ldt_StopDDN terminates the DDN server.
3. ldt_DeleteFiles handles MACS requests to delete file sets.
4. ldt_RetainFiles handles MACS requests to mark file sets for retention.
5. ldt_RcvDDN watches for incoming DDNs and parses and processes them when they arrive from EDC DAAC.
6. ldt_RsndSuspDANs, triggered by the enabling by the MACS of DAN transfer, finds all file sets for which DANs have been suspended, updates them to indicate that they no longer have suspended DANs, and calls ldt_SendDAN to resend each of the associated DANs.

The LDTS design contains highly modular, cohesive modules that can easily be updated and maintained. Existing software is integrated into the design where applicable.

11.3.2 Detailed Module Design

11.3.2.1 ldt_SendDAN

The MACS will invoke ldt_SendDAN (Figure 11–3) to send a DAN. The MACS first creates a LDTS_File_Set_Info database record with the contact sequence identifier and file version identifier for the contact period file set that will be transferred to EDC DAAC. The MACS invokes ldt_SendDAN, passing as arguments the contact sequence identifier and file version identifier pair that identifies the output file set.

On operator request, the MACS also will invoke ldt_SendDAN to resend a failed DAN. To resend, the MACS simply invokes ldt_SendDAN, passing as arguments the contact sequence identifier and file version identifier of the file set whose DAN is to be resent, and an indication that a RESEND is requested.

ldt_SendDAN retrieves the associated filenames from the LPS_File_Info database table and verifies that all of the files in the file set do indeed exist and are readable. After the communication link to EDC DAAC is established, ldt_SendDAN begins to retrieve all filenames and file types to build a DAN for the contact period file set. The DAN_Transfer_State in the database is checked. If DAN transfer is disabled, ldt_SendDAN marks the file set as having a suspended DAN and logs this in the LPS Journal. Otherwise, a DAN for the given file set is transmitted over a socket to EDC DAAC. For a RESEND, the DAN will include only those file groups that have not been transferred, that is, those file groups with failed transfer status in the LDT_File_Group_Info database table.

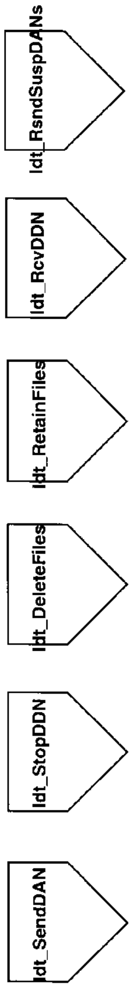
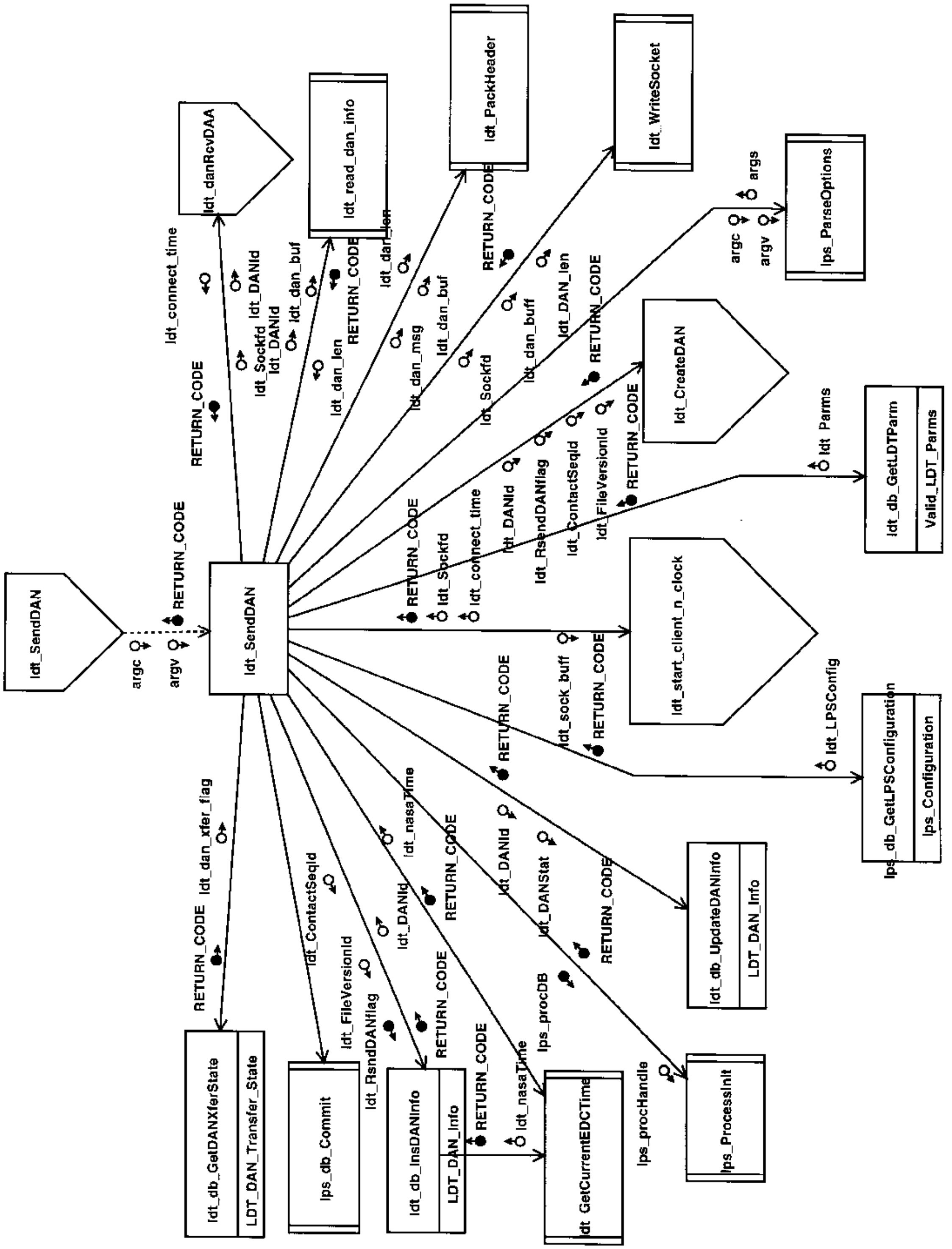


Figure 11–2. LDTS Top-Level Structure Chart

Figure 11–3. ldt_SendDAN Structure Chart



After a DAN is sent to EDC DAAC, ldt_SendDAN waits for a DAA message from EDC DAAC to acknowledge LPS receipt of the DAN and any DAN errors. If a DAA in response to a DAN is received, ldt_SendDAN verifies the DAN errors and records the DAN status into the LDT_DAN_Info, LDT_File_Set_Info, and LDT_File_Group_Info database tables.

Finally, the events of sending a DAN and receiving a DAA are recorded in the LPS Journal file.

11.3.2.2 ldt_StopDDN

ldt_StopDDN (Figure 11–4) is invoked by the MACS to terminate the DDN server. It first checks the DDN server's processing stage in the LDT_DDN_Server_Info database table. If a DDN is under, ldt_StopDDN waits until the DDN processing is complete and terminates the DDN server.

11.3.2.3 ldt_DeleteFiles

ldt_DeleteFiles (Figure 11–5) is called with a contact sequence identifier and file version identifier pair to identify a file set and an optional argument. When the optional "only delete ingested files" argument is present, ldt_DeleteFiles will only delete the files in file groups that have successfully transferred. Otherwise, all the files in the contact period file set will be deleted. Both the MACS and ldt_RcvDDN invoke ldt_DeleteFiles. If invoked by ldt_RcvDDN (the third optional argument is not allowed), the contact period file set is deleted automatically unless it is marked for retention. When the MACS invokes ldt_DeleteFiles, the MACS can specify whether to delete a partial file set or a whole file set by the presence of the optional argument.

11.3.2.4 ldt_RetainFiles

ldt_RetainFiles (Figure 11–6) is implemented by the MACS via the user interface to mark a file set for retention. Once a file set is marked for retention, it cannot be deleted automatically except by the MACS.

11.3.2.5 ldt_RcvDDN

ldt_RcvDDN (Figure 11–7) is forked by the MACS at LPS startup and runs until the MACS shuts it down. It first creates the socket server to accept one, and only one, connection from EDC DAAC. After a connection is established, it waits for the DDN message from EDC DAAC. On receipt of a DDN, ldt_RcvDDN checks the data transfer status for the file set, file groups, and individual files and records the status into the LDT_File_Set_Info, LDT_File_Group_Info, and LPS_File_Info tables. It also generates a short DDA message and sends over the socket to EDC DAAC. ldt_RcvDDN periodically sends a health message to the operator.

Finally, the events of receiving a DDN and sending a DDA are recorded in the LPS Journal file.

11.3.2.6 ldt_RsndSuspDANs

ldt_RsndSuspDANs (Figure 11–8) is called when DAN transfer is enabled. It retrieves all suspended DANs from the LDT_DAN_Info database table and creates a list of contact sequence identifier and file version identifier pairs, identifying file sets that have suspended DANs associated with them. ldt_RsndSuspDANs then invokes ldt_SendDAN to resend the DANs with this list.

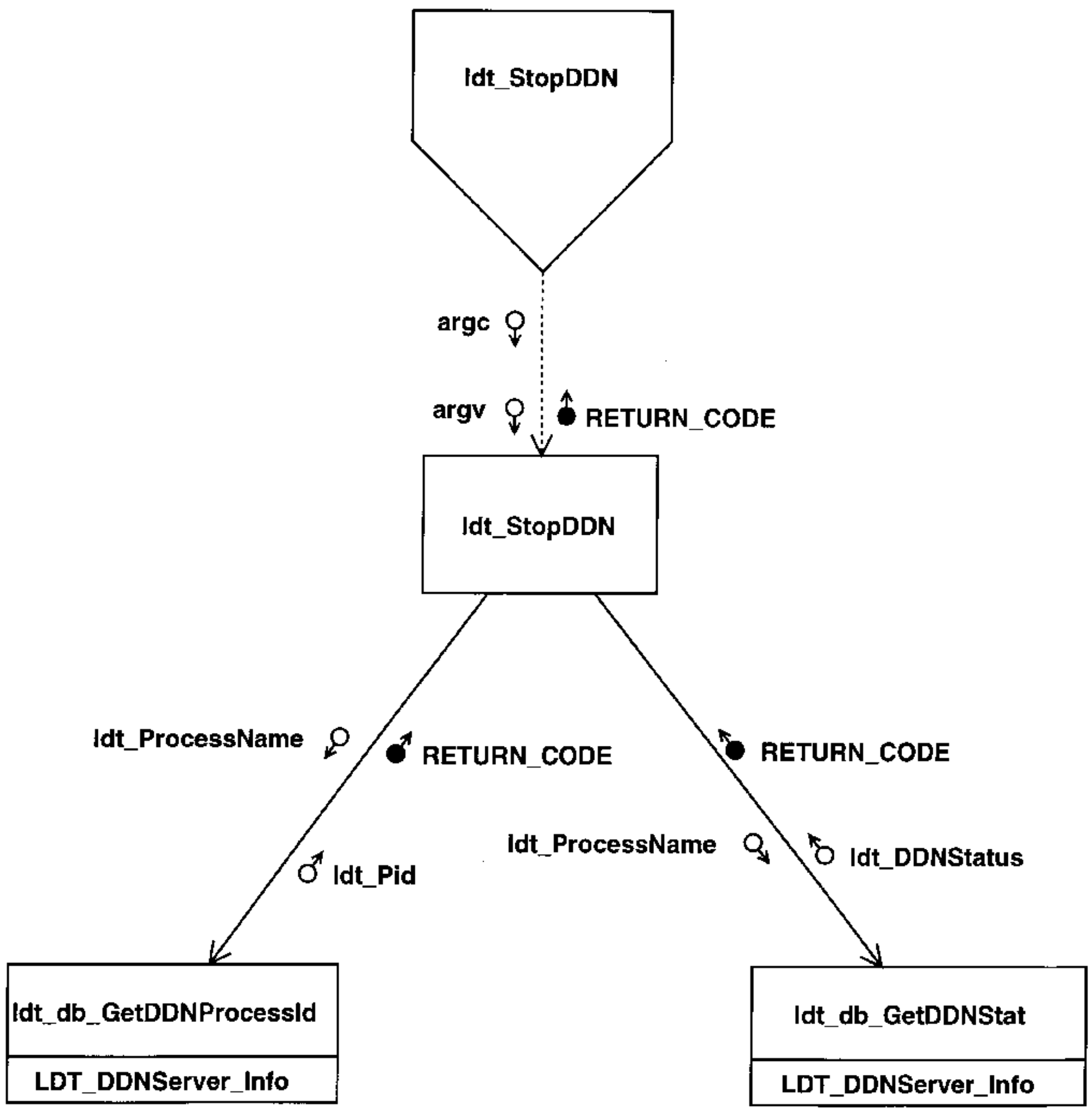


Figure 11–4. ldt_StopDDN Structure Chart

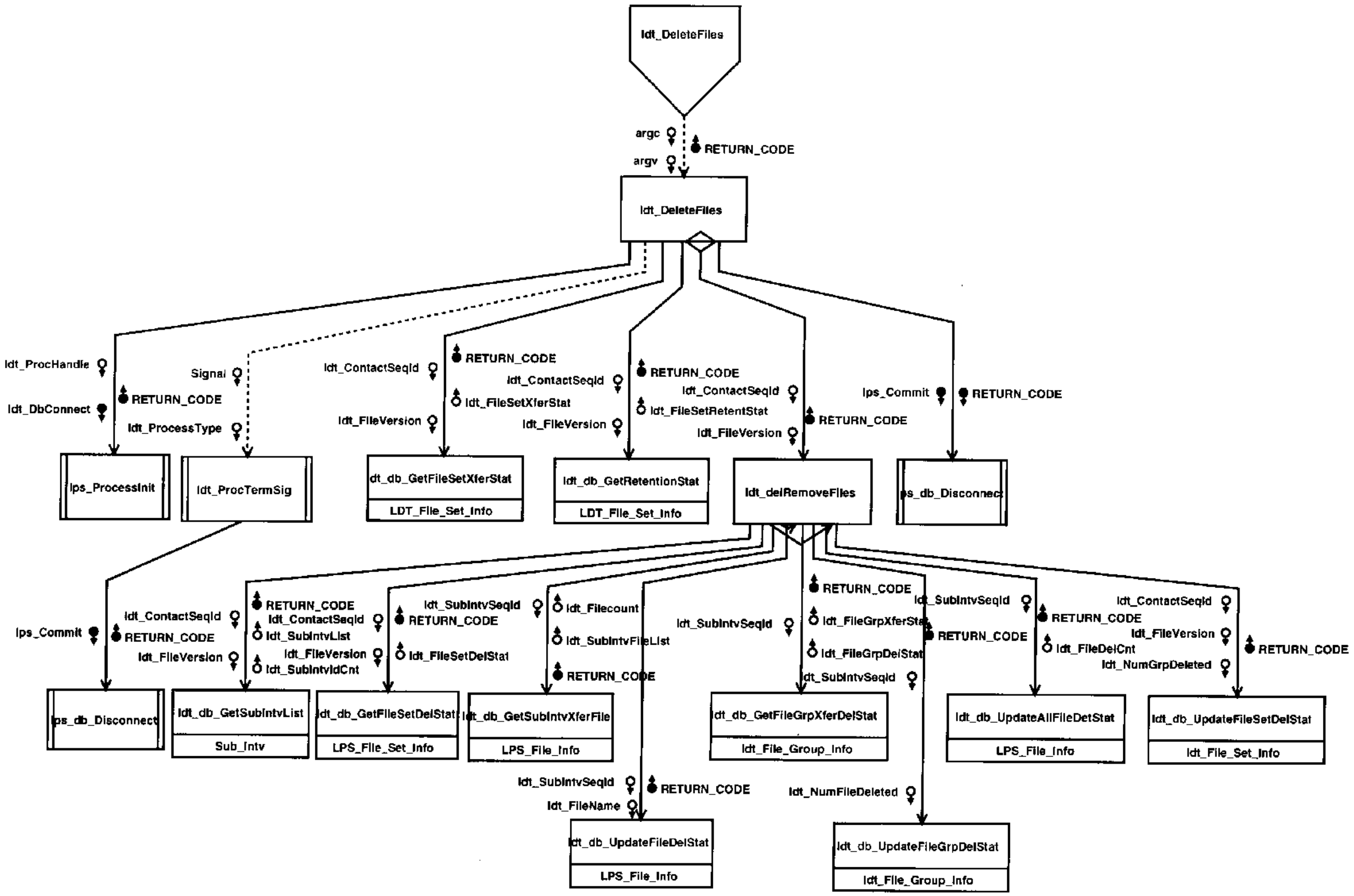


Figure 11–5. ldt_DeleteFiles Structure Chart

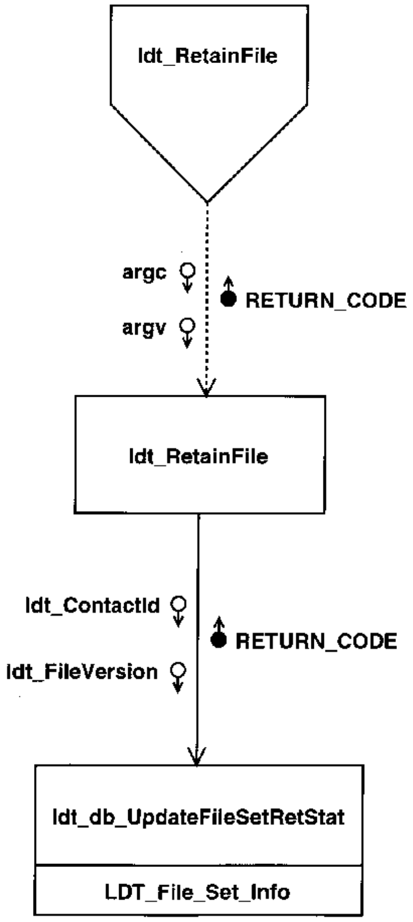
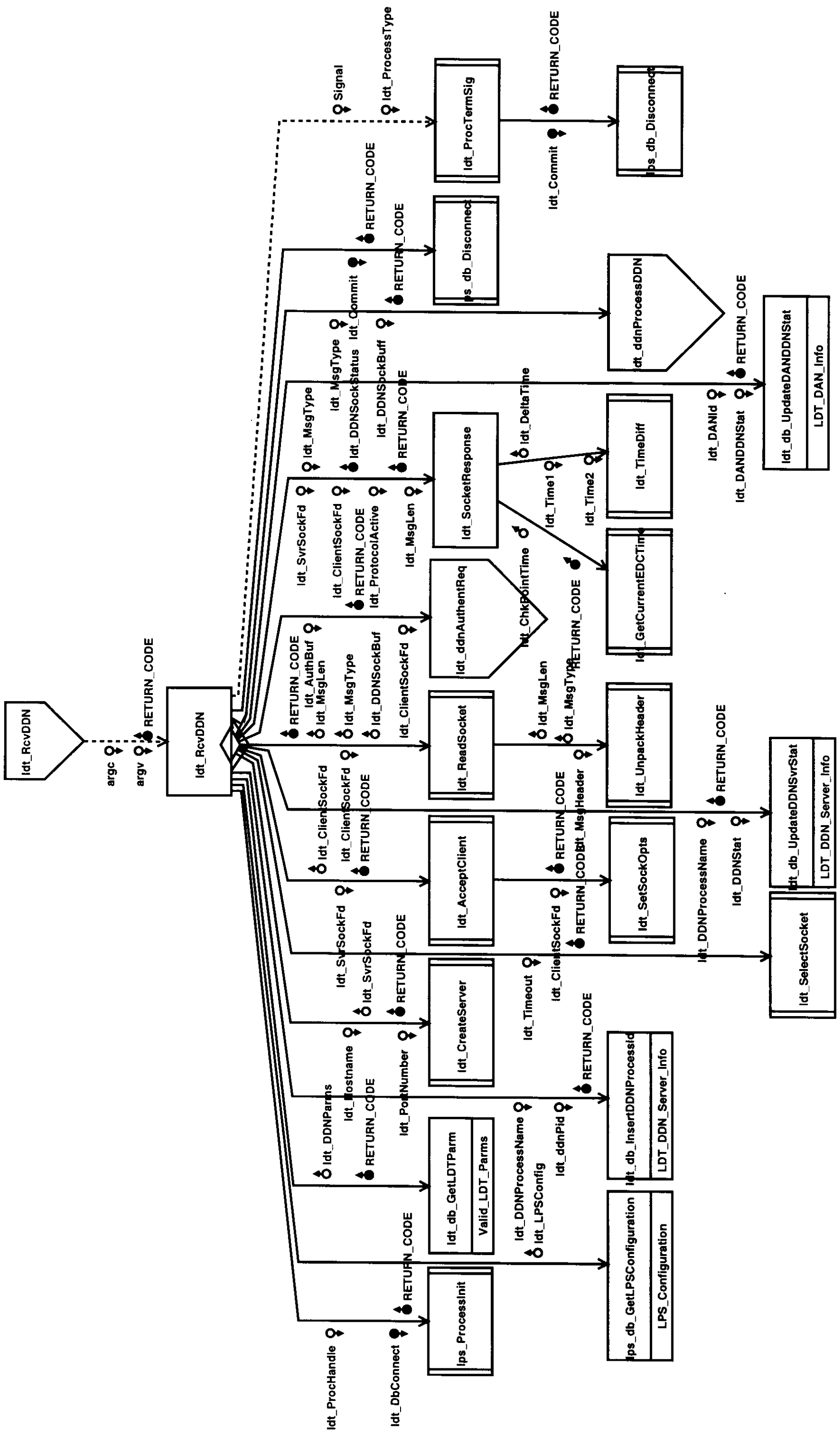


Figure 11–6. ldt_RetainFiles Structure Chart

Figure 11–7. Idt_RcvDDN Structure Chart


ldt_RcvDDN
argc
argv
RETURN_CODE
ldt_RcvDDN
lps_ProcessInit
ldt_ProcHandle
ldt_DbConnect
RETURN_CODE
lps_db_GetLPSConfiguration
LPS_Configuration
ldt_DDNProcessName
ldt_LPSConfig
ldt_db_GetLDTParm
Valid_LDT_Parms
ldt_DDNParms
RETURN_CODE
ldt_db_InsertDDNProcessId
LDT_DDN_Server_Info
ldt_ddnPid
RETURN_CODE
ldt_CreateServer
ldt_Hostname
ldt_PortNumber
ldt_SvrSockFd
RETURN_CODE
ldt_AcceptClient
ldt_SvrSockFd
ldt_ClientSockFd
ldt_ClientSockFd
RETURN_CODE
ldt_SetSockOpts
ldt_Timeout
ldt_ClientSockFd
RETURN_CODE
RETURN_CODE
ldt_SelectSocket
ldt_db_UpdateDDNSvrStat
LDT_DDN_Server_Info
ldt_DDNProcessName
ldt_DDNStat
RETURN_CODE
ldt_ReadSocket
RETURN_CODE
ldt_AuthBuf
ldt_MsgLen
ldt_MsgType
ldt_DDNSockBuf
ldt_ClientSockFd
ldt_UnpackHeader
ldt_MsgLen
ldt_MsgType
ldt_MsgHeader
ldt_ddnAuthentReq
ldt_ChkPointTime
RETURN_CODE
ldt_GetCurrentEDCTime
ldt_SocketResponse
ldt_SvrSockFd
ldt_ClientSockFd
RETURN_CODE
ldt_ProtocolActive
ldt_MsgLen
ldt_MsgType
ldt_DDNSockStatus
ldt_DDNSockBuf
RETURN_CODE
ldt_TimeDiff
ldt_Time1
ldt_Time2
ldt_DeltaTime
ldt_db_UpdateDANDDNStat
LDT_DAN_Info
ldt_DANId
ldt_DANDDNStat
RETURN_CODE
ldt_ddnProcessDDN
ldt_MsgType
ldt_Commit
RETURN_CODE
lps_db_Disconnect
RETURN_CODE
ldt_ProcTermSig
Signal
ldt_ProcessType
lps_db_Disconnect
ldt_Commit
RETURN_CODE

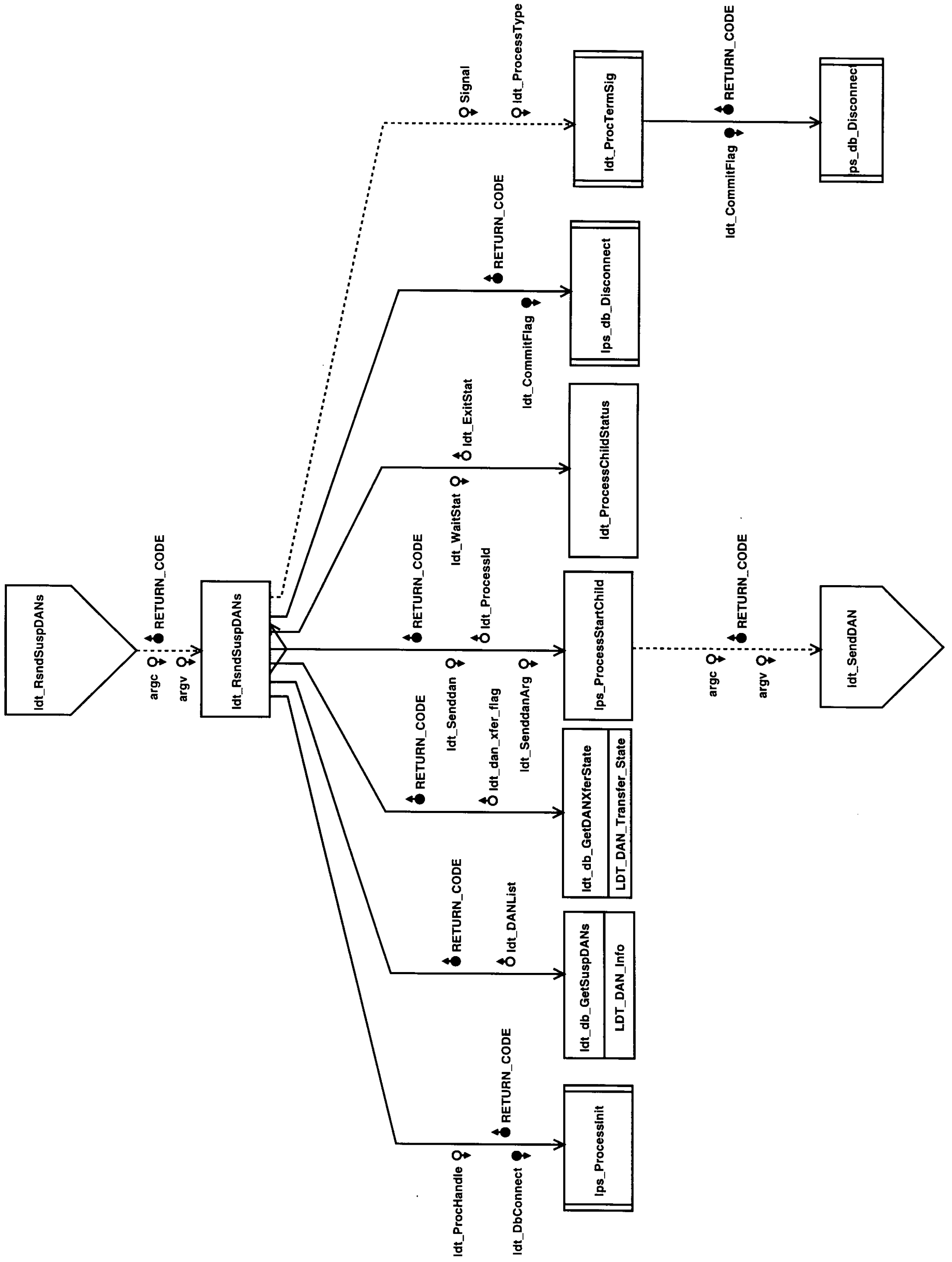


Figure 11–8. ldt_RsndSuspDANs Structure Chart

Section 12. Database Design

12.1 Purpose and Scope

The LPS database provides support for keeping track of Q&A information of Landsat 7 wideband data received from LGS. It also provides operational support. This section describes the LPS database logical design and physical design.

During the preliminary design phase, a logical database design was established incorporating the functional data requirements and the interface between the database and the LPS subsystems. For the detailed design phase, the logical database was refined and expanded as required to support the detailed system design. The principal focus of this phase is the physical database design.

12.2 Logical Design

The goal of logical design is to establish the functional design of the database in terms of entities, relationships, and attributes. The logical design is independent of any particular physical implementation. The logical database design is used as the foundation for the physical database design.

12.2.1 Entity Relationship Model

The LPS entity relationship (ER) model is a conceptual representation of the data for the LPS application. The ER model is expressed in terms of entities in the LPS environment, the relationships or associations among those entities, and the properties of the entities and their relationships. Entities represent data items that play a functional role in the LPS application and have their own set of attributes. Table 12–1 contains a description of each LPS entity identified during the conceptual design process. The attributes of each entity are analyzed and described in Table 12–2.

The LPS ER model is expressed as the entity relationship diagram (ERD), which is a graphical representation of the ER model as illustrated in Figure 12–1. In addition to entities described previously, relationships are also included in the diagram. The relationships represent the association between the instances of one or more entities that are of interest to the LPS. Each relationship has a cardinality associated with it that describes the number of instances of one entity that can be associated with each instance of the entity to which it relates. For example, each subinterval (Sub_Intv) has more than one Level 0R output file (LPS_File_Info), but only one MFPS accounting information (MFP_Acct).

12.2.2 Integrity Constraints

Integrity constraints provide a means of ensuring that changes made to the database do not result in a loss of data consistency. Thus, integrity constraints guard against accidental damage to the database. At the design stage of the LPS database, the following types of integrity constraints are considered:

- Domain Rules – Define a set of legal values (called domain rules) for a column or set of columns and instruct the DBMS to check all values that users try to store and reject any that are illegal. The domains and ranges are defined in Table 12–2.

- Unique Keys—Prevent storing more than one record having duplicate values in a specified column or set of columns in a table by defining an implicit unique index on each set of columns by the DBMS. The DBMS will then reject any attempt to store duplicate values for columns that should be unique.

- Primary Keys—Prevent storing more than one record having duplicate values in a specified column or set of columns in a table. It also disallows null values entering in the specified columns. This constraint is enforced by the implicit creation of a unique index on that column and a not-null constraint on that column by the DBMS. (The primary keys are defined in Table 12–2.)

- Foreign Keys (referential constraints)—Whenever a primary or alternate key value from one table is stored as a foreign key in another table, that the reference is valid must be ensured. To preserve referential integrity, each time a record of a table containing foreign keys is inserted, it must be checked. The DBMS should check that the new foreign key values actually exist in the other table and should not allow the insertion or update if they do not exist. Also, the DBMS should check every time a record or a table is deleted or modified because records of other tables may exist that point to that table row using foreign keys. (Foreign keys are defined in Table 12–2.)

- Null—A rule defined on a single column that allows or disallows inserts or updates of rows containing a null for the column (specified in Table 12–2).

12.2.3 Logical Schema

The logical schemata in Table 12–2 represents the normalized tables (in BCNF). It also includes domain rules. The primary keys are identified by asterisks (*) and the foreign keys by plus signs (+).

Table 12–1. LPS Entity Descriptions (1 of 2)

Entity Name	Entity Description
Band_Gain_States	Initial gain setting for each band and each scene; may also indicate that gain changes later in scene
Bands_Present	Information about bands present on a PCD cycle basis
Contact_Schedules	Set of contact periods containing start and stop times when Landsat 7 spacecraft downlinks wideband data to LGS; schedule is coming from LGS in a hardcopy form
IDP_Acct	Aggregate scene accounting information for IDPS, including ACCA scores
LDT_DAN_Info	DAN status information
LDT_DAN_Transfer_State	Indication of whether of not DAN transfer is enabled
LDT_DDN_Server_Info	Status information about DDN server
LDT_File_Group_Info	File group status information on subinterval basis
LDT_File_Set_Info	File set status information on contact period basis
LPS_File_Info	Level 0R files status information
LPS_Configuration	Set of parameters used to configure LPS; some parameters are used when system starts up, while others are used during processing

Table 12–1. LPS Entity Descriptions (2 of 2)

Entity Name	Entity Description
MFP_Acct	Aggregate accounting information from MFPS that includes Level 0R Q&A information on subinterval basis
MFP_MJF_Acct	Aggregate of Level 0R accounting information from MFPS on major frame basis
PCD_Acct	Aggregate accounting information from PCDS that includes processed PCD Q&A information on subinterval basis
PCD_MJF_Acct	Aggregate of PCD Q&A information on PCDS major frame basis
PCD_Scene_Acct	Aggregate scene accounting information on contact basis
Process_Id	Parent process identifier of an LPS child process
Processing_Version_Info	Data processing count information per contact period
RDC_Acct	Raw data capture accounting information on contact basis
RDP_Acct	Aggregate accounting information from RDPS that includes return-link Q&A information on contact basis
Sub_Intv	List of subinterval information belonging to a contact period generated by MFPS
UTC_UT1_Corrections	Universal time coordinated (UTC) and UT1 thresholds in seconds for PCDS processing
Valid_Band_Parms	Aggregate of information that includes parameters for Browse and Band subprocesses, as well as parameters for color associations, reduction, compression, and contrast stretching
Valid_CCSDS_Parms	List of current and history CCSDS parameters that controls CCSDS frame synchronization and bit slip correction
Valid_Detector_Gains_Bias	Individual detector gain and bias adjustments used for radiometric correction algorithm
Valid_LDT_Parms	File transfer parameters for LDTS
Valid_MFP_Parms	Validated MFPS setup parameters
Valid_MFP_Thres	Validated MFPS threshold values
Valid_MWD_Parms	Band color associations and X11 options used by MWD
Valid_PCD_Parms	Validated PCD parameters used in processing PCD
Valid_PCD_Thres	Validated PCD threshold parameters used in processing PCD
Valid_RDP_Thres	Validated RDP processing thresholds
Valid_Scene_Parms	Validated general mission information and parameters provided by IAS and used to calculate longitude, latitude, WRS scene identifier, and Sun elevation and azimuth
Valid_Sensor_Align_Parms	Sensor pixel alignment information
Valid_WRS_Parms	Validated WRS table containing information for each WRS scene

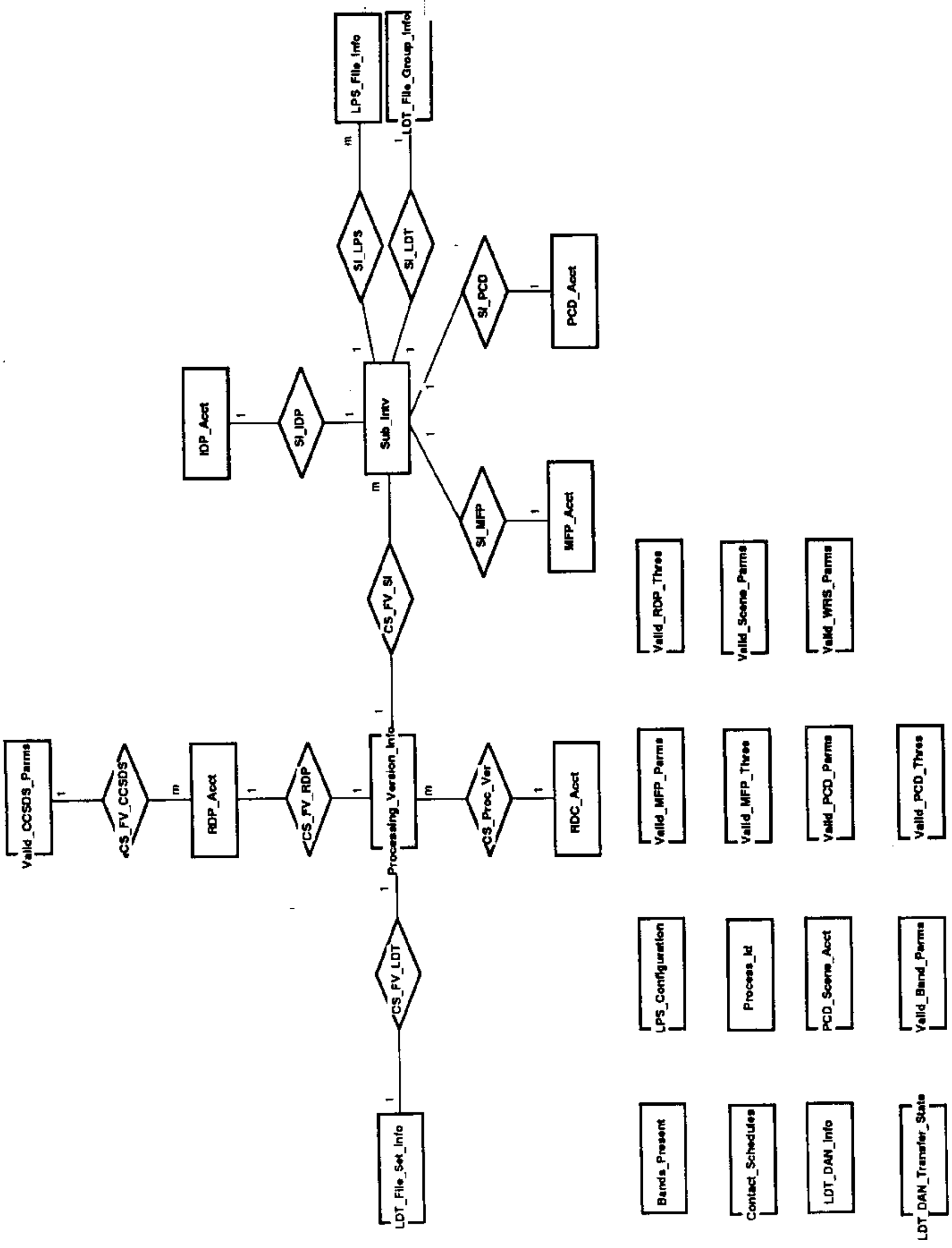


Figure 12–1. LPS ER Diagram

Table 12–2. LPS Database Schema (1 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
Bands_ Present	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval period sequence
	PCD_Cycle_ Time *	SC_Time_St ring	Varchar2 (30)	(not null)	PCD cycle time (1/16th of msec)
	Band_Present	Band_Byte	Char (1)	(not null)	Bands present Fmt1:1,2,3,4,5,6 Fmt2: 6,7,8
Band_Gain_ States	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval period sequence
	Scene_Center_ Time *	SC_Time_ String	Varchar2 (30)	(not null)	Actual WRS scene center time when image was taken (1/16th of msec)
	Band *	Band_Id_Int	Number (1)	1<=X<=8 (not null)	Band number
	Band_Gain	Band_Gain_ Char	char (1)	L, H (not null)	Band gain at start of a scene: L = low gain H = high gain
	Band_Gain_ Change	Band_Gain_ Change_ Char	char (1)	0, +, – (not null)	Band gain change within a scene: 0 = no change + = low to high – = high to low
	Band_Gain_ Scan_Num	Band_Gain_ Int	Number (6)	0 <= X <= 12000 (null)	Scan line number for first band gain change
Contact_ Schedules	Scheduled_ Sequence_Id *	Id_Int	Number (10)	X > 0 (not null)	Key for scheduled contact
	Scheduled_ Start_Time	Ground_ Time_String	Date	(not null)	Scheduled AOS (sec)
	Scheduled_Sto p_Time	Ground_ Time_String	Date	(not null)	Scheduled LOS (sec)

Table 12–2. LPS Database Schema (2 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
IDP_Acct	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval period sequence
	Scene_Center_ Time	SC_Time_ String	Varchar2 (30)	(not null)	WRS scene center time when image was taken (1/16th of msec)
	Scene_Start_ Time	SC_Time_ String	Varchar2 (30)	(null)	Scene start time calculated by PCDS
	Scene_Stop_ Time	SC_Time_ String	Varchar2 (30)	(null)	Scene stop time calculated by PCDS
	Sub_Intv_ Scene_Number *	Scene_Id_In t	Number (2)	0<=X<= 99 (not null)	Scene number within subinterval
	Scene_Center_ Scan_Number	Count_Int	Number (5)	-175 <=X <= 11725 (not null)	Scan number or major frame number whose center is closest to center of nominal WRS scene
	CCA_ Quadrant1_Scor e	Cloud_ Percentage_ Real	Number (5,2)	0.00 <= X <= 100.00	Cloud covered percentage of first quadrant of a scene
	CCA_ Quadrant2_Scor e	Cloud_ Percentage_ Real	Number (5,2)	0.00 <= X <= 100.00	Cloud covered percentage of second quadrant of a scene
	CCA_ Quadrant3_Scor e	Cloud_ Percentage_ Real	Number (5,2)	0.00 <= X <= 100.00	Cloud covered percentage of third quadrant of a scene
	CCA_ Quadrant4_Scor e	Cloud_ Percentage_ Real	Number (5,2)	0.00 <= X <= 100.00	Cloud covered percentage of fourth quadrant of a scene
	CCA_Aggregate _Score	Cloud_ Percentage_ Real	Number (5,2)	0.00 <= X <= 100.00	Percentage of scene that is cloud covered

Table 12–2. LPS Database Schema (3 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Full_Partial_ Scene	Scene_Flag _Int	Char (1)	F, P	Flag indicating that associated scene is full or partial: F = full scene P = partial scene
LDT_DAN_ Info	DAN_ Sequence_Num *	Id_Int	Number (10)	X > 0 (not null)	DAN identifier
	Contact_ Sequence_Id +	Id_Int	Number (10)	X > 0 (not null)	Identifies contact period sequence
	File_Version_ Number +	Count_Int	Number (10)	0 <=X <= 9 (not null)	Identifies version of processed data for given contact
	DAN_Ack_Time _Stamp	Time_Int	Number (12)	(not null)	DAA receipt time in sec (GMT)
	DAN_Status	Status_Int	Number (1)	0 <= X <= 6 (not null)	DAN processing status: 0 = send wait (default) 1 = suspended 2 = send success 3 = send error 4 = DAA received 5 = DAA timeout 6 = canceled
	Num_Sub_ Intervals	Count_Int	Number (2)	0 <= X <= 99 (not null)	Number of subintervals in contact period
	DDN_Receive_ Time_Stamp	Ground_ Time_String	Number (12)	(null)	DDN receive time in sec (GMT)
	DDN_Status	Status_Int	Number (1)	0 <= X <= 3 (not null)	0 = DDN waiting 1 = DDN processing 2 = DDN process success 3 = DDN process failed

Table 12–2. LPS Database Schema (4 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
LDT_DAN_ Transfer_ State	Xfer_State	State_Int	Number (1)	X = 0,1 (not null)	Indicates if DAN are currently being sent from LDTS to EDC DAAC: 0 = disabled 1 = enabled
LDT_DDN_ Server_Info	Process_Name	Name_Strin g	Varchar2 (10)	(not null)	DDN server process name
	Process_Id *	Id_Int	Number (10)	X >= 0 (not null)	DDN server process identifier
	Processing_ Stage	Status_Int	Number (1)	0 <= X <= 5	0 = idle 1 = receiving DDN 2 = verifying DDN 3 = deleting files 4 = creating DDA 5 = sending DDA
LDT_File_ Group_Info	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Subinterval identifier used to identify file group
	Number_Of_File s	Count_Int	Number (4)	X >= 0 (not null)	Total number of files in current file group
	File_Group_ DAA_Disposition	Status_Strin g	Varchar2 (40)	(not null)	File group DAA disposition (status)
	File_Group_ Xfer_Status	Status_Int	Number (1)	X = 0,1 (not null)	File group transfer status: 1 = error 0 = success (default)
	File_Group_ Deletion_Status	Status_Int	Number (1)	X = 0,1,2 (not null)	File group deletion status: 0 = not deleted (default) 1 = partially deleted 2 = deleted

Table 12–2. LPS Database Schema (5 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
LDT_File_ Set_Info	Contact_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies contact period sequence
	File_Version_ Number *+	Count_Int	Number (10)	0 <=X<=9 (not null)	Identifies version of processed data for given contact
	Creation_Time	Ground_ Time_String	Date	(null)	DAN creation time (default = null)
	Retention_Statu s	Status_Int	Number (1)	X = 0,1 (not null)	File set retention status: 0 = no retention (default) 1 = marked for retention
	Num_Of_File_ Groups	Count_Int	Number (4)	X >= 0 (not null)	Number of file groups in contact period (default = 0)
	Num_Of_File_ Groups_DAA_ Failed	Count_Int	Number (4)	X >= 0 (not null)	Number of file groups DAA failed (default = 0)
	File_Set_ Deletion_Status	Status_Int	number (1)	X = 0,1,2 (not null)	0 = not deleted (default) 1 = partially deleted 2 = deleted
	Num_Of_File_ Groups_Deleted	Count_Int	number (4)	X >= 0 (not null)	Number of file groups deleted
	Num_Of_File_ Groups_Xfer_ Failed	Count_Int	Number (4)	X >= 0 (not null)	Number of file groups transfer failed
	File_Set_Xfer_ Status	Status_Int	Number (1)	0 <= X <= 4 (not null)	File set transfer status: 0 = success 1 = failure 2 = wait DDN 3 = partial 4 = Level 0R in progress (default)

Table 12–2. LPS Database Schema (6 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	File_Set_ Transfer_ Disposition	Status_Int	Number (4)	X >= 0 (not null)	File set transfer disposition (default = 9999)
	File_Set_DAA_ Status	Status_Int	Number (1)	0 <= X <= 4 (not null)	File set DAA status: 0 = success: 1 = error 2 = partial 3 = timeout 4 = unknown (default)
	File_Set_DAA_ Disposition	Status_Strin g	Varchar2 (40)	(null)	File set DAA disposition (status) (default = null)
LPS_ Configuratio n	LPS_Hardware_ String_Id	Id_String	Varchar2 (20)	(not null)	LPS source host string identification
	ECS_Hardware_ String_Id	Id_String	Varchar2 (20)	(not null)	ECS destination host string identification
	Capture_Source	Id_String	Varchar2 (3)	(not null)	Capture source
	Spacecraft_Id	Id_String	Varchar2 (8)	(not null)	Landsat 7 spacecraft identifier
	Instrument_Id	Id_String	Varchar2 (4)	(not null)	Instrument identifier (ETM+)
	LPS_Software_ Ver_Num	Version_ Num_String	Varchar2 (5)	(not null)	LPS software version number
	LPS_User_Id	Id_String	Varchar (20)	(not null)	ECS user identifier at LPS machine
	LPS_Password	Id_String	Varchar (20)	(not null)	Encrypted password
	ECS_User_Id	Id_String	Varchar (20)	(not null)	LPS user identifier at ECS machine
	ECS_Password	Id_String	Varchar (20)	(not null)	Encrypted password
	LPS_Port_Num	Id_Int	Number (4)	(not null)	LPS host port number
	ECS_Port_Num	Id_Int	Number (4)	(not null)	ECS host port number

Table 12–2. LPS Database Schema (7 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	IAS_Param_File_Ver_Num	Version_ String	Varchar2 (4)	(not null)	IAS parameter file version number
LPS_File_ Info	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies file's subinterval
	File_Path	File_Path_ String	Varchar2 (256)	(not null)	Output file directory
	File_Name *	File_Name_ String	Varchar2 (256)	(not null)	Output filename
	File_Type	File_Type_ String	Varchar2 (13)	(not null)	Output file type: CAL = calibration MTA = metadata PCD = PCD MSD = MSCD BX0 = 1 <= X <= 7 (band 1 = 7) B8X = 1 <= X <= 4 (band 8, segment X) RXX = 01 <= XX <= 99 (browse/ scene XX)
	File_Deletion_ Status	Status_Int	Number (1)	0 <= X <= 3 (not null)	Output file deletion status: 0 = exist (default) 1 = deleted
	File_Xfer_ Disposition	Status_Int	Number (6)	X >= 0 (null)	Output file transfer status
	File_Xfer_Time_Stamp	File_Transfer_Time_String	Varchar2 (30)	(null)	Output file transfer time stamp in character string
MFP_Acct	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval for which Q&A appears
	Mjf_CADU_ Rcvd_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of received non-fill CADUs
	Mjf_CADU_Fly_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of flywheel CADUs

Table 12–2. LPS Database Schema (8 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Mjf_CADU_ Inverted_ Polarity_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of CADUs received with inverted polarity
	Mjf_CADU_ Polarity_ Change_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of changes to the CADU polarity
	Mjf_CADU_Bit_ Slip_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of CADUs with bit slips
	Mjf_CADU_ Sync_Err_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of CADUs with sync errors
	Mjf_CADU_ Missing_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of missing CADUs
	Mjf_CADU_RS_ Corr_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of correctable VCDU headers
	Mjf_CADU_RS_ Uncorr_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of uncorrectable VCDU header
	Mjf_CADU_ BCH_Data_Corr _Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of CADUs with corrected BCH errors in mission data zone
	Mjf_CADU_ BCH_Data_ Uncorr_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of CADUs with uncorrected BCH errors in mission data zone
	Mjf_CADU_ Data_Bits_Corr_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Total number of BCH corrected bits in mission data zone
	Mjf_CADU_ BCH_Ptr_Corr_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of CADUs with corrected BCH errors in data pointer zone
	Mjf_CADU_ BCH_Ptr_ Uncorr_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of CADUs with uncorrected BCH errors in data pointer zone
	Mjf_CADU_ BCH_Ptr_Bits_ Corr_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Total number of BCH corrected bits in data pointer zone

Table 12–2. LPS Database Schema (9 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Mjf_CADU_ CRC_Err_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of CADUs with CRC errors
	Mjf_Cnt	Count_Int	Number (8)	11725 >= X >= 0 (not null)	Count of major frames (ETM+ scans)
	Mjf_Time_Code _Err_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of imagery timecode errors
	Mjf_Full_Fill_Cn t	Count_Int	Number (8)	X >= 0 (not null)	Count of entirely filled ETM+ major frames
	Mjf_Part_Fill_C nt	Count_Int	Number (8)	X >= 0 (not null)	Count of partially filled ETM+ major frames
MFP_MJF_ Acct	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval for which Q&A appears
	MFP_MJF_Time *	SC_Time_ String	Varchar2 (30)	(not null)	Major frame time (1/16th of msec) measured by satellite
	CADUs_ Received	Count_Int	Number (6)	0 <= X <= 999999 (not null)	Number of non-fill CADUs for this MJF
	Fly_Wheel_ CADUs	Count_Int	Number (6)	0 <= X <= 999999 (not null)	Number of flywheel CADUs for this major frame
	RS_Error_ VCDUs	Count_Int	Number (6)	0 <= X <= 999999 (not null)	Number of RS errors for this major frame
	BCH_Corr_ VCDUs	Count_Int	Number (6)	0 <= X <= 999999 (not null)	Number of BCH corrected CADUs for this major frame
	BCH_Uncorr_ VCDUs	Count_Int	Number (6)	0 <= X <= 999999 (not null)	Number of uncorrectable CADUs for this major frame
	Num_Bit_Errors	Count_Int	Number (8)	0 <= X <= 99999999 (not null)	Bit errors in this major frame

Table 12–2. LPS Database Schema (10 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	ETM_Timecode_Error_Flag	Timecode_ Error_Flag_ Int	Number (1)	X = 0,1 (not null)	Timecode error in this major frame: 0 = false 1 = true
	MJF_Filled_Flag	Fill_Flag_ Char	Char (1)	E, P, N (not null)	Major frame fill: E = entirely filled P = partially filled N = not filled
PCD_Acct	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval for which Q&A appears
	Num_PCD_MJF	Count_Int	Number (3)	0 <= X <= 255 (not null)	Number of PCD major frames
	First_PCD_MJF_ Time	SC_Time_ String	Varchar2 (30)	(not null)	First PCD major frame time (1/16th of msec) in a subinterval
	PCD_Stop_Time	SC_Time_ String	Varchar2 (30)	(not null)	Last PCD major frame time (1/16th of msec) in a subinterval
	ETM_Last_On_ Time	SC_Time_ String	Number (16,7)	(not null)	Last instrument on time (msec)
	ETM_Last_Off_ Time	SC_Time_ String	Number (16,7)	(not null)	Last instrument off time (msec)
	UT1_Corrections	UT1_Real	Number (6,5)	-0.90000 <= X <= 0.90000	UTC–UT1 in seconds from IAS parameter file
PCD_MJF_ Acct	Sub_Intv_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval for which Q&A appears
	PCD_MJF_Time *	SC_Time_ String	Varchar2(30)	(not null)	PCD major frame time (1/16th second)

	PCD_Words_ Received	Count_Int	Number (6)	0 <= X <= 999999 (not null)	Total number of PCD words, extracted from unpacked PCD words for this PCD major frame

Table 12–2. LPS Database Schema (11 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Total_PCD_MNF	Count_Int	Number (3)	0 <= X <= 999 (not null)	Total number of PCD minor frames constructed for this PCD major frame
	Failed_PCD_ Votes	Count_Int	Number (6)	0 <= X <= 999999 (not null)	Number of PCD words that encountered byte-voting errors during packing for this PCD major frame
	Num_PCD_ MNF_Sync_ Errors	Count_Int	Number (3)	0 <= X <= 999 (not null)	Number of PCD minor frames with sync errors
	Num_PCD_ Filled_MNF	Count_Int	Number (3)	0 <= X <= 999 (not null)	Number of PCD filled minor frames
	PCD_Filled_ MJF_Flag	Flag_Int	Char (1)	E, P, N (not null)	PCS major frame fill: E = entirely filled P = partially filled N = not filled
	PCD_ADP_MJF _Flag	Flag_Char	Char(1)	G, R, M (not null)	Availability of spacecraft attitude data points received and processed for this PCD major frame G = good R = rejected M = missing
	PCD_EDP_MJF _Flag	Flag_Char	Char (1)	G, R, M (not null)	Availability of spacecraft ephemeris data points received and processed for this PCD major frame G = good R = rejected M = missing

Table 12–2. LPS Database Schema (12 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
PCD_Scene_Acct	Sub_Intv_Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval for which Q&A appears
	Scene_Center_Time *	SC_Time_String	Varchar2 (30)	(not null)	WRS scene center time when image is taken (1/16th of msec)
	WRS_Path_Nominal	WRS_Path_Int	Number (3)	1 <= X <= 233 (not null)	Nominal scene center (WRS path is east to west index of WRS table)
	WRS_Row_Nominal	WRS_Row_Int	Number (3)	1 <= X <= 248 (not null)	Nominal scene center (WRS path is north to south index of WRS table)
	Horizontal_Display_Shift	Display_Shift_Real	Number (8,4)	-9999.00 00 <= X <= 9999.0000(not null)	Horizontal display shift of WRS scene
	Sun_Azimuth	Sun_Real	Number (10,7)	-180.000 0 000 <= X <= 180.0000000 (not null)	Solar angle
	Sun_Elevation	Sun_Real	Number (10,7)	-90.000 0 000 <= X <= 90.0000000 (not null)	Solar elevation
	Cal_Door_Activity_Status	Cal_Int	Number (1)	X = 0 or 1 (not null)	Calibration door activity for given period of time 0 = closed 1 = open
	Scene_Center_Lat	Lat_Real	Number (7,4)	-90.0000 <= X <= 90.0000 (not null)	Latitude of scene center
	Scene_Center_Lon	Lon_Real	Number (7,4)	-180.000 0 <= X <= 180.0000 (not null)	Longitude of scene center

Table 12–2. LPS Database Schema (13 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Scene_Upper_ Left_Lat	Lat_Real	Number (7,4)	-90.0000 <= X <= 90.0000 (not null)	Latitude of upper-left scene corner
	Scene_Upper_ Left_Lon	Lon_Real	Number (7,4)	-180.000 0 <= X <= 180.0000 (not null)	Longitude of upper-left scene corner
	Scene_Upper_ Right_Lat	Lat_Real	Number (7,4)	-90.0000 <= X <= 90.0000 (not null)	Latitude of upper-right scene corner
	Scene_Upper_ Right_Lon	Lon_Real	Number (7,4)	-180.000 0 <= X <= 180.0000 (not null)	Longitude of upper-right scene corner
	Scene_Lower_ Left_Lat	Lat_Real	Number (7,4)	-90.0000 <= X <= 90.0000 (not null)	Latitude of lower-left scene corner
	Scene_Lower_ Left_Lon	Lon_Real	Number (7,4)	-180.000 0 <= X <= 180.0000 (not null)	Longitude of lower-left scene corner
	Scene_Lower_ Right_Lat	Lat_Real	Number (7,4)	-90.0000 <= X <= 90.0000 (not null)	Latitude of lower-right scene corner
	Scene_Lower_ Right_Lon	Lon_Real	Number (7,4)	-180.000 0 <= X <= 180.0000 (not null)	Longitude of lower-right scene corner
Process_Id	Contact_ Sequence_Id +	Id_Int	Number (10)	X > 0 (null)	Identifies contact period sequence
	File_Version_ Number +	Count_Int	Number (10)	0 <= X <= 9 (not null)	Data processing count
	Parent_Process_ Id +	Id_Int	Number (10)	X > 0 (not null)	MACS parent process identifier of Level 0R processing
	Task_Name	Task_Name_ String	Varchar2 (25)	(null)	Level 0R processing task

Processing_Version_Info	Contact_Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies contact period sequence

Table 12–2. LPS Database Schema (14 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	File_Version_ Number *	Count_Int	Number (10)	0 <= X <= 9 (not null)	Data processing count
	Level0R_ Completion_Flag	Boolean	Number (1)	X = 1, 0 (not null)	Indicator of Level 0R process in progress 1 = Yes, 0 = No
RDC_Acct	Contact_ Sequence_Id *	Id_Int	Number (10)	X > 0 (not null)	Identifies contact period sequence
	LPS_Hardware_ String_Id	Id_String	Varchar2 (20)	(not null)	LPS string identifier that captures data
	Capture_Source	Id_String	Varchar2 (3)	(null)	Contact channel identifier
	Scheduled_ Start_Time	Ground_ Time_String	Date	(null)	Scheduled AOS (sec)
	Scheduled_Stop_ Time	Ground_ Time_String	Date	(null)	Scheduled LOS (sec)
	Actual_Start_ Time	Ground_ Time_String	Date	(not null)	Actual data acquisition start time (sec)
	Actual_Stop_ Time	Ground_ Time_String	Date	(not null)	Actual data acquisition end time (sec)
	Raw_Data_File_ Name	File_Name_ String	Varchar2 (512)	(not null)	Raw data filename including path
	Received_Data_ Vol	Data_Size_ Real	Number (7,2)	X >= 0.00 (not null)	Received data volume (megabytes)
	Expected_Data_ Vol	Data_Size_ Real	Number (7,2)	X >= 0.00 (not null)	Data volume estimation between actual start and stop time (megabytes per sec)
	Scheduled_ Data_Vol	Data_Size_ Real	Number (7,2)	X >= 0.00 (not null)	Data volume estimation between scheduled start and stop time (megabytes)

Table 12–2. LPS Database Schema (15 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	File_Size	File_Size_ Real	Number (7,2)	X >= 0.00 (not null)	File size in megabytes
	Transmission_ Rate	Rate_Real	Number (7,2)	X >= 0.00 (not null)	Received data volume and actual start and stop time difference in megabytes
	Isolate_Flag	Boolean	Number (1)	X = 1,0 (not null)	Indicates if data capture had dedicated process
	Suspend_Flag	Boolean	Number (1)	X = 1,0 (not null)	Level 0R processing suspension indicator
	On_Line_Flag	Boolean	Number (1)	X = 1,0 (not null)	File indicator 0 = offline 1 = online
	Archive_Flag	Boolean	Number (1)	X = 1,0 (not null)	File archive to disk indicator 1 = archived 0 = not archived
RDP_Acct	Contact_ Sequence_Id *+	Id_Int	Number (10)	X > 0 (not null)	Identifies contact period sequence
	File_Version_ Number *+	Count_Int	Number (10)	0 <= X <= 9 (not null)	Data processing count
	CCSDS_Parms_ Id +	Id_Int	Number (10)	X > 0 (not null)	Valid_CCSDS_ Parms table's record identifier
	Inverted_CADU_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of CADUs received with inverted polarity
	Polarity_ Change_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of polarity changes on CADUs
	CADU_Bit_Slip_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of CADUs that have bit slips
	CADU_Sync_ Err_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of CADUs with sync errors

Table 12–2. LPS Database Schema (16 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	CADU_Rcv_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of received CADUs
	CADU_Flywheel _Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of flywheel CADUs
	Fill_CADU_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of number of fill CADUs
	CADU_CRC_ Err_Cnt	Count_Int	Number (8)	X >= 0 (not null)	CRC errors when CCSDS processing raw wideband data set
	VCDU_Header1 _Corr_Err_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of RS-correctable VCDU headers for VCID1
	VCDU_Header2 _Corr_Err_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of RS-correctable VCDU headers for VCID2
	VCDU_Header_ Uncorr_Err_Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of uncorrectable VCDU headers
	BCH_Data_ Corrected_Err_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of CADUs with BCH errs corrected for mission data zone in VCDU
	BCH_Data_ Uncorrected_Err _Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of CADUs with BCH errors uncorrected for mission data zone in VCDU
	BCH_Data_ Corrected_Bits_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of bits corrected by BCH in mission data area and BCH mission zone
	BCH_Ptr_ Corrected_Err_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of correctable BCH pointer field errors encountered when CCSDS was processing raw wideband data set

Table 12–2. LPS Database Schema (17 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	BCH_Ptr_ Uncorrected_Err _Cnt	Count_Int	Number (8)	X >= 0 (not null)	Number of uncorrectable BCH pointer field errors encountered
	BCH_Ptr_ Corrected_Bits_ Cnt	Count_Int	Number (8)	X >= 0 (not null)	Count of bits corrected by BCH in data pointer area and BCH pointer zone
Sub_Intv	Contact_ Sequence_Id +	Id_Int	Number (10)	X > 0 (not null)	Identifies contact period sequence
	File_Version_ Number +	Count_Int	Number (10)	0 <= X <= 9 (not null)	Data reprocessing count
	Sub_Intv_ Sequence_Id +	Id_Int	Number (10)	X > 0 (not null)	Identifies subinterval period sequence
	MF_Start_Time	SC_Time_ String	Varchar2 (30)	(not null)	Subinterval start time (1/16th of msec)
	MF_Stop_Time	SC_Time_ String	Varchar2 (30)	(null)	Subinterval stop time (1/16th of msec)
	VCID	VCID_Int	Number (1)	X = 1, 2 (not null)	Virtual channel identification
	Sub_Intv_ Number	Sub_Count_ Int	Number (2)	0 <= X <= 99 (not null)	Subinterval number within contact
UTC_UT1_ Corrections	Correction_date *	Date_Int	Number (5)	(not null)	UTC–UT1 correction date (modified Julian date)
	UT1_Corrections	UT1_Real	Number (6,5)	-0.90000 <= X <= 0.90000 (not null)	UTC–UT1 in seconds from IAS parameter file
Valid_Band _Parms	Multi1	Band_Int	Number (1)	1 <= X <= 6 (not null)	Band parameter specifying first of three bands to process for browse (format 1)

Table 12–2. LPS Database Schema (18 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Multi2	Band_Int	Number (1)	1 <= X <= 6 (not null)	Band parameter specifying second of three bands to process for browse (format 1)
	Multi3	Band_Int	Number (1)	1 <= X <= 6 (not null)	Band parameter specifying third of three bands to process for browse (format 1)
	Subs	Reduction_ Int	Number (3)	X >= 0 (not null)	Subsampling reduction ratios (format 1)
	Wave	Reduction_ Int	Number (3)	X >= 0 (not null)	Wavelet iteration
	JPEG_Quality	Reduction_ Int	Number (3)	0 <= X <= 100 (not null)	JPEG quality factor
	Contrast_ Stretch_Factor	Str_Fac_Int	Number (3)	0 <= X <= 49 (not null)	Clipping factor used in the contrast stretching histogram
Valid_ CCSDS_ Parms	CCSDS_Parms_ Id *	Id_int	Number (10)	X > 0 (not null)	Valid_CCSDS_ Parms table's record identifier
	Insertion_Time	Ground_ Time_String	Date	(not null)	Record insertion time
	CADU_Search_ Tol	Tolerance_ Int	Number (1)	1 <= X <= 3 (not null)	Search tolerance parameter
	CADU_Check_ Tol	Tolerance_ Int	Number (1)	0 <= X <= 3 (not null)	Check tolerance parameter
	CADU_Flywheel _ Tol	Tolerance_ Int	Number (1)	0 <= X <= 3 (not null)	Flywheel tolerance parameter
	CADU_Sync_ Mark_Check_ Error_Tol	Tolerance_ Int	Number (1)	0 <= X <= 3 (not null)	Check error tolerance parameter
	CADU_Sync_ Lock_Error_Tol	Tolerance_ Int	Number (1)	0 <= X <= 3 (not null)	Lock error tolerance parameter
	CADU_Bit_Slip_ Corr_Extent	Tolerance_ Int	Number (1)	0 <= X <= 3 (not null)	Slip correction extent parameter

Table 12–2. LPS Database Schema (19 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Xfer_Frame_Trouble_File	Boolean	Number (1)	X = 0 or 1 (not null)	Trouble file generation flag 1 = yes 0 = no
	BCH_Flag	Boolean	Number (1)	X = 0,1 (not null)	BCH error checking flag 1 = yes 0 = no
Valid_ Detector_ Gain_Bias	Band	Band_Int	Number (2)	1 <= X <= 6 (not null)	Only for format 1 band (1 = 6)
	Detector	Det_Int	Number (2)	1 <= X <= 16 (not null)	Bands 1–5, 16 detectors Band 6, eight detectors
	High_Gain	Gain_Real	Number (8,4)	(not null)	High gain
	High_Gain_Bias	Bias_Real	Number (8,4)	(not null)	High gain bias
	Low_Gain	Gain_Real	Number (8,4)	(not null)	Low gain
	Low_Gain_Bias	Bias_Real	Number (8,4)	(not null)	Low gain bias
Valid_LDT_ Parms	Num_Auth_ Request	Count_Int	Number (4)	X >= 0 (not null)	Maximum logon requests
	Timeout_Auth_ Request	Time_Int	Number (4)	X >= 0 (not null)	Timeout value for authentication request in seconds
	Num_Send_ DAN_Attempt	Count_Int	Number (4)	X >= 0 (not null)	Number of sending DAN attempts
	Timeout_Send_ DAN_Attempt	Time_Int	Number (4)	X >= 0 (not null)	Timeout value for sending DAN attempts in seconds
	Timeout_ Receive_DAA	Time_Int	Number (4)	X >= 0 (not null)	Timeout value for receiving DAA in seconds
	Timeout_ Receive_DDN	Time_Int	Number (4)	X >= 0 (not null)	Timeout value for receiving DDN in seconds
	Num_Send_ DDA_Attempt	Count_Int	Number (4)	X >= 0 (not null)	Number of sending DDA attempts

Table 12–2. LPS Database Schema (20 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Timeout_Send_ DDA	Time_Int	Number (4)	X >= 0 (not null)	Timeout value for sending DDA in seconds
	Read_Sleep_ Second	Time_Int	Number (10)	X >= 0 (not null)	Sleep time between socket reads
	LDT_ECS_Msg_ Save	Boolean	Number (1)	X = 0,1 (not null)	Message save flag 1 = saved 0 = not saved
Valid_MFP_ Parms	Fill_Value	Fill_Val_Int	Number (6)	(not null)	Fill value for major frame processing
	Sub_Intv_Delta	Delta_Real	Number (10,7))	X > 0.00 (not null)	Subinterval determining delta
	Mjf_Data_Perio d	Rate_Real	Number (8,7)	X > 0.00 (not null)	Data rate threshold
	Time_Range_T ol	Tolerance_ Real	Number (8,7)	X > 0.00 (not null)	Time range tolerance
	Maj_Vote_Tol	Tolerance_ Int	Number (8)	X >= 0 (not null)	Majority voting tolerance
	Max_Time_Spa n	Tolerance_ Int	Number (8)	X >= 0 (not null)	Timespan of major frame expected
	Eol_Tol	Tolerance_ Int	Number (8)	X >= 0 (not null)	Search zone around nominal location of EOL
	Mjf_Sync_Tol	Tolerance_ Int	Number (8)	X >= 0 (not null)	Number of bit errors allowed
	Mjf_Sync_Size	Tolerance_ Int	Number (8)	X >= 0 (not null)	Minimum number of bytes in sync
	Max_Mnf_ Counter	Tolerance_ Int	Number (8)	X >= 0 (not null)	Maximum minor frame counter
	ETM_Plus_ Trouble_File	Boolean	Number (1)	X = 0,1 (not null)	Trouble file generation indicator 1 = yes 0 = no

	Min_Mjf_Count_Per_Sub_Intv	Tolerance_Int	Number (8)	X >= 0 (not null)	Minimum number of scans (major frames) in a subinterval compared to keep output files

Table 12–2. LPS Database Schema (21 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
Valid_MFP_ Thres	Mjf_CADU_Seq _Err_Thres	Threshold_ Int	Number (8)	X >= 0 (not null)	Threshold for number of sequence counter errors
	Mjf_Sync_Thres	Threshold_ Int	Number (8)	X >= 0 (not null)	Maximum number of sync errors allowed
	Mnf_Ctr_Thres	Threshold_ Int	Number (8)	X >= 0 (not null)	Maximum number of minor frame errors allowed
	Eol_Thres	Threshold_ Int	Number (8)	X >= 0 (not null)	Maximum number of EOL errors allowed
	Tc_Thres	Threshold_ Int	Number (8)	X >= 0 (not null)	Maximum number of timecode errors allowed
	Full_Mjf_Thres	Threshold_ Int	Number (8)	X >= 0 (not null)	Maximum number of full filled major frames allowed
	Part_Mjf_Thres	Threshold_ Int	Number (8)	X >= 0 (not null)	Maximum number of partial filled major frames allowed
Valid_MWD _ Parms	Fmt1_Red_Ban d	Band_Int	Number (1)	1 <= X <= 6 (not null)	Band and color coordination for MWD
	Fmt1_Green_ Band	Band_Int	Number (1)	1 <= X <= 6 (not null)	Band and color coordination for MWD
	Fmt1_Blue_Ban d	Band_Int	Number (1)	1 <= X <= 6 (not null)	Band and color coordination for MWD
	Fmt2_Red_Ban d	Band_Int	Number (1)	6 <= X <= 8 (not null)	Band and color coordination for MWD
	Fmt2_Green_ Band	Band_Int	Number (1)	6 <= X <= 8 (not null)	Band and color coordination for MWD
	Fmt2_Blue_Ban d	Band_Int	Number (1)	6 <= X <= 8 (not null)	Band and color coordination for MWD

Table 12–2. LPS Database Schema (22 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Xoptions	X_String	Varchar2 (256)	(null)	X Windows system option
Valid_PCD_ Parms	PCD_Frame_Fill	Fill_Value_ Int	Number (8)	X >= 0 (not null)	Value to fill missing PCD
Valid_PCD_ Thres	Ephem_Position _Upper	Ephem_Pos _Real	Number (9,2)	-8.3886 x 1000 <= X <= 8.3886 x 1000 (not null)	Largest valid ephemeris position data point (km)
	Ephem_Position _Lower	Ephem_Pos _Real	Number (9,2)	-8.3886 x 1000 <= X <= 8.3886 x 1000 (not null)	Smallest valid ephemeris position data point (km)
	Ephem_Velocity _Upper	Ephem_Vel _Real	Number (3,2)	-8 <= X <= 8 (not null)	Largest valid ephemeris velocity data point (km per second)
	Ephem_Velocity _Lower	Ephem_Vel _Real	Number (3, 2)	-8 <= X <= 8 (not null)	Smallest valid ephemeris velocity data point (km per second)
	Ephem_ Crossproduct_ Max	Ephem_Vel _Real	Number (7,2)	X=53200.00 (not null)	Largest valid crossproduct of ephemeris velocity and position
	Ephem_ Crossproduct_ Min	Ephem_Vel _Real	Number (7,2)	X=53000.00 (not null)	Smallest valid crossproduct of ephemeris velocity and position
	Attitude_ Quaternion_Tol	Att_Tol_Rea l	Number (6,2)	X >= 0.0 (not null)	Tolerance used to check attitude quaternion
	Num_Missing_ Data_ Words	Threshold_I nt	Number (8)	X >= 0 (not null)	Threshold for reporting errors when PCD information words are missing

Table 12–2. LPS Database Schema (23 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Num_Failed_ Votes	Threshold_I nt	Number (8)	X > =0 (not null)	Threshold for reporting errors when unsuccessful majority votes are performed
Valid_RDP_ Thres	Sync_Thres	Threshold_I nt	Number (8)	X >= 0 (not null)	Maximum number of CADUs with sync errors allowed before notifying operator
	CRC_Thres	Threshold_I nt	Number (8)	X >= 0 (not null)	Maximum number of CADUs with CRC errors allowed before notifying operator
	RS_Thres	Threshold_I nt	Number (8)	X >= 0 (not null)	Maximum number of CADUs with RS errors allowed before notifying operator
	BCH_Thres	Threshold_I nt	Number (8)	X >= 0 (not null)	Maximum number of CADUs with BCH errors allowed before notifying operator
Valid_ Scene_ Parms	ETM_Body_ Trans_Matrix_11	ETM_Real	Number (3,2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 11 from ETM+ line of sight at center of scan to spacecraft body
	ETM_Body_ Trans_Matrix_12	ETM_Real	Number (3,2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 12
	ETM_Body_Tra ns_ Matrix_13	ETM_Real	Number (3, 2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 13
	ETM_Body_ Trans_Matrix_21	ETM_Real	Number (3,2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 21

Table 12–2. LPS Database Schema (24 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	ETM_Body_ Trans_Matrix_22	ETM_Real	Number (3,2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 22
	ETM_Body_ Trans_Matrix_23	ETM_Real	Number (3,2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 23
	ETM_Body_ Trans_Matrix_31	ETM_Real	Number (3,2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 31
	ETM_Body_ Trans_Matrix_32	ETM_Real	Number (3,2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 32
	ETM_Body_ Trans_Matrix_33	ETM_Real	Number (3,2)	-1.00 <= X <= 1.00 (not null)	Transformation matrix cell 33
	Semi_Major_Axis	Axis_Real	Number (10,3)	0.00 <= X (not null)	Distance from apogee or perigee to center of orbit ellipse
	Semi_Minor_Axis	Axis_Real	Number (10,3)	0.00 <= X (not null)	Polar axis radius
Valid_ Sensor_ Align_Parms	Sensor_Align_ Title *	Align_Id_ String	Varchar2 (40)	(not null)	Pixel alignment title
	Sensor_Align_ Value	Align_Val_Real	Number (8,5)	(not null)	Pixel alignment value
Valid_WRS_ Parms	WRS_Path_ Nominal *	WRS_Row_ Int	Number (3)	0 <= X <= 233 (not null)	WRS path number
	WRS_Row_ Nominal *	WRS_Path_ Int	Number (3)	0 <= X <= 248 (not null)	WRS row number
	Scene_Center_ Latitude	Lat_Lon_ Real	Number (10,7)	-90.00 <= X <= 90.00 (not null)	Scene center latitude
	Scene_Center_ Longitude	Lat_Lon_ Real	Number (10,7)	-180.00 <= X <= 180.00 (not null)	Scene center longitude
	Trailing_Left_ Latitude	Lat_Lon_ Real	Number (10,7)	-90.00 <= X <= 90.00 (not null)	Upper-left corner latitude
	Trailing_Left_ Longitude	Lat_Lon_ Real	Number (10,7)	-180.00 <= X <= 180.00 (not null)	Upper-left corner longitude

Table 12–2. LPS Database Schema (25 of 25)

Entity Name	Attribute Name	LPS Domain Name	ORACLE Data Type	Range/ Value	Attribute Description
	Trailing_Right_ Latitude	Lat_Lon_ Real	Number (10,7)	-90.00 <= X <= 90.00 (not null)	Upper-right corner latitude
	Trailing_Right_ Longitude	Lat_Lon_ Real	Number (10,7)	-180.00 <= X <= 180.00 (not null)	Upper-right corner longitude
	Leading_Left_ Latitude	Lat_Lon_ Real	Number (10,7)	-90.00 <= X <= 90.00 (not null)	Lower-left corner latitude
	Leading_Left_ Longitude	Lat_Lon_ Real	Number (10,7)	-180.00 <= X <= 180.00 (not null)	Lower-left corner longitude
	Leading_Right_ Latitude	Lat_Lon_ Real	Number (10,7)	-90.00 <= X <= 90.00 (not null)	Lower-right corner latitude
	Leading_Right_ Longitude	Lat_Lon_ Real	Number (10,7)	-180.00 <= X <= 180.00 (not null)	Lower-right corner longitude

12.3 Data Usage Analysis

Data usage analysis examines the manner in which the LPS subsystems interact with the database tables and the frequency of those interactions. From this evaluation, denormalization of the logical design and other implementation options may be identified to improve performance.

12.3.1 Functional Usage Analysis

The functional usage of the database is presented in a create, retrieve, update, and delete (CRUD) matrix (Table 12–3), which depicts the interactions between LPS subsystems and database schemata. In the table, the convention used is as follows: I = insert, U = update, D = delete, and Q = query.

12.3.2 Performance Analysis

Because all of the LPS subsystems interact with the database, the performance requirements for LPS subsystems include database performance considerations. Main factors affecting the database performance were considered and are discussed in the following sections.

12.3.2.1 Denormalizations

Denormalization is a process in which columns belonging to one table are redundantly defined in another table to reduce or eliminate the need to query the original table. Thus, expensive join

Table 12–3. LPS Subsystem CRUD Matrix

Schema\Subsystem	RDCS	RDPS	MFPS	PCDS	IDPS	MACS	LDTS
Bands_Present				I		D,Q	
Bands_Gain_States					I	Q	
Contact_Schedules	D					I,D,Q	
IDP_Acct					I	Q	
LDT_DAN_Info							I,D,Q
LDT_DAN_Transfer_State						I,U,Q	Q
LDT_DDN_Server_Info							I,D
LDT_File_Group_Info							I,Q
LDT_File_Set_Info							I,U,Q
LPS_File_Info			I	I	I	I,Q	U,Q
LPS_Configuration			Q	Q	Q	I,U,Q	
MFP_Acct			I			Q	
MFP_MJF_Acct			I			D,Q	
PCD_Acct				I		Q	
PCD_MJF_Acct				I		D,Q	
PCD_Scene_Acct				I		Q	
Process_Id						I,D,Q	I
Processing_Version_Info	I					I,Q	
RDC_Acct	I	Q				Q	
RDP_Acct		I	Q			Q	
Sub_Intv			I,Q	Q	Q	Q	Q
UTC_UT1_Corrections				Q		I,U,Q	
Valid_Band_Parms					Q	I,U,Q	
Valid_CCSDS_Parms		Q				I,Q	
Valid_Detector_Gain_Bands					Q	I,U,Q	
Valid_LDT_Parms						I,U,Q	Q
Valid_MFP_Parms			Q			I,U,Q	
Valid_MFP_Thres			Q			I,U,Q	
Valid_MWD_Parms					Q	I,U,Q	
Valid_PCD_Parms				Q		I,U,Q	
Valid_PCD_Thres				Q		I,U,Q	
Valid_RDP_Thres		Q				I,U,Q	
Valid_Scene_Parms				Q		I,U,Q	
Valid_Sensor_Align_Parms			Q			I,U,Q	
Valid_WRS_Parms				Q		I,Q	

operations can be avoided. In determining which tables should be denormalized, the following considerations were weighed:

- Would the improved retrieval performance produce a sufficient overall improvement in throughput processing to outweigh the costs of additional insertions and updates?
- Would retrieval processing during critical times be sufficiently improved to outweigh the costs of storing redundant data?
- Would the changes create an unacceptable impact on maintaining the data integrity?

The logical schema in Table 12–2 are in BCNF after having gone through the denormalization process.

12.3.2.2 Indexes

In the LPS database, no index is created explicitly. However, ORACLE implicitly creates an index for each primary key constraint.

12.4 Physical Design

Physical database design is the process of converting logical data design into a design that can be supported by the type of DBMS selected. Physical database design has several objectives:

- To optimize the logical schema as necessary while preserving the consistency and flexibility of data
- To determine the data storage requirements
- To define data integrity and security rules
- To optimize the usage of the DBMS capabilities

The LPS physical schema information resulted from the incorporation of these objectives as described in Table 12–2. The LPS database physical schema is based on the assumption that it will be implemented using the ORACLE relational DBMS.

12.4.1 Tables, Columns, and Indexes

The tables and columns compose the basic physical database design units. A database table is a basic unit of data storage. Actual data is stored in rows according to predefined table schema. Entities and relationships become tables, and attributes become table columns.

As part of the data performance analysis, indexes were determined for the tables. Indexes must be chosen carefully because they require maintenance as tuples are added and deleted from the tables and because they require extra storage. Because indexes can significantly reduce the data retrieval time, they are crucial for performance. Indexes can also be used to ensure that data in a column or combination of columns is always unique within a given table.

12.4.2 Database Storage Requirements

The data in the LPS database is divided into three different categories for estimation of storage requirements:

- **Static data** seldom changes. Verified parametric data and threshold data belong to this category.
- **Dynamic data** changes often, but is not retained for long periods of time. Contact schedule data belongs to this class.
- **History data** is infrequently modified, but is continuously being added to history tables. Q&A data belongs to this category.

For the static data tables, the number of rows is constant. The storage needed for the static data table is the sum of each table's row length times the number of rows in the table. The row length includes DBMS overhead for each column and row in the tables. In addition, the DBMS organizes the data into blocks requiring additional table overhead. For the dynamic data tables, storage is calculated based on the maximum number of rows expected at a time. For the history data tables, the number of rows is computed based on the daily record numbers and the data retention days.

In addition to the storage for data, storage is required for indexes and for DBMS overhead files. The considerations for indexes are similar to the considerations for tables.

Table 12–4 lists the number of rows expected for each table and the corresponding number of bytes required. In the table, the following constraint name conventions are used:

Primary key:	pk_xxx	(attr1, attr2, ..., attrn)
Not null:	nn_xxx	attr
Check constraint:	ck_xxx	attr between value x and y
Foreign key:	fk_xxx	referencing table_name (attr1, att2, ..., attrn)

Table 12–4. LPS Database Table Constraints (1 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
Rdc_Acct	nn_ra_cont_seq_id	Contact_Sequence_Id
	pk_rdc_acct	Contact_Sequence_Id
	ck_ra_cont_seq_id	(0 < Contact_Sequence_Id)
	nn_ra_lps_hw_string_id	LPS_Hardware_String_Id
	nn_ra_actual_stat_time	Actual_Start_Time
	nn_ra_actual_stop_time	Actual_Stop_Time
	nn_ra_raw_data_file_name	Raw_Data_File_Name
	nn_ra_recvd_data_vol	Received_Data_Vol

Table 12–4. LPS Database Table Constraints (2 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	ck_ra_recvd_data_vol	(0.0 <= Received_Data_Vol)
	nn_ra_exp_data_vol	Expected_Data_Vol
	ck_ra_exp_data_vol	(0.0 <= Expected_Data_Vol)
	nn_ra_sched_data_vol	Scheduled_Data_Vol
	ck_ra_sched_data_vol	(0.0 <= Scheduled_Data_Vol)
	nn_ra_file_size	File_Size
	ck_ra_file_size	(0.0 <= File_Size)
	nn_ra_transm_rate	Transmission_Rate
	nn_ra_isolate_flag	Isolate_Flag
	nn_ra_suspend_flag	Suspend_Flag
	nn_ra_on_line_flag	On_Line_Flag
	nn_ra_archive_flag	Archive_Flag
	ck_ra_archive_flag	(Archive_Flag between 0 and 1)
Processing_Version_Info	nn_pvi_contact_seq_id	Contact_Sequence_Id
	fk_Processing_Version_Info	Rdc_Acct (Contact_Sequence_Id)
	ck_pvi_contact_seq_id	(0 < Contact_Sequence_Id)
	nn_pvi_file_ver_num	File_Version_Number
	ck_pvi_file_ver_num	(0 <= File_Version_Number)
	nn_pvi_l0r_comp_flag	L0r_Completion_Flag
	ck_pvi_l0r_comp_flag	(L0r_Completion_Flag between 0 and 1)
	pk_processing_version_info	(Contact_Sequence_Id, File_Version_Number)
Sub_Intv	nn_si_cont_seq_id	Contact_Sequence_Id
	ck_si_cont_seq_id	(0 < Contact_Sequence_Id)
	nn_si_file_ver_num	File_Version_Number
	ck_si_file_ver_num	(0 <= File_Version_Number)
	nn_si_sub_int_seq_id	Sub_Intv_Sequence_Id
	pk_Sub_Intv	Sub_Intv_Sequence_Id
	ck_si_sub_int_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_si_mf_start_time	MF_Start_Time
	nn_si_mf_stop_time	MF_Stop_Time
	nn_si_vcid	VCID
	ck_si_vcid	(VCID between 1 and 2)
	nn_si_sub_intv_num	Sub_Intv_Number

Table 12–4. LPS Database Table Constraints (3 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	ck_si_sub_intv_num	(Sub_Intv_Number between 0 and 99)
	fk_Sub_Intv	processing_version_info (Contact_Sequence_Id, File_Version_Number)
Bands_Present	nn_bp_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_Bands_Present	Sub_Intv (Sub_Intv_Sequence_Id)
	ck_bp_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_bp_pcd_cycle_time	PCD_Cycle_Time
	nn_bp_band_present	Band_Present
	pk_bands_present	(Sub_Intv_Sequence_Id, PCD_Cycle_Time)
Band_Gain_States	nn_bgs_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_Band_Gain_States	Sub_Intv (Sub_Intv_Sequence_Id)
	ck_bgs_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_bgs_scene_center_time	Scene_Center_Time
	nn_bgs_band	Band
	ck_bgs_band	(Band between 1 and 8)
	nn_bgs_band_gain	Band_Gain
	ck_bgs_band_gain	(Band_Gain = L or H)
	nn_bgs_band_gain_change	Band_Gain_Change
	ck_bgs_band_gain_change	(Band_Gain_Change = 0, +, or –
	ck_bgs_band_gain_scan_num	(Band_Gain_Scan_Num between 0 and 12000)
	pk_Band_Gain_States	(Sub_Intv_Sequence_Id, Scene_Center_Time, Band
Contact_Schedules	nn_cs_sched_seq_id	Scheduled_Sequence_Id
	ck_cs_sched_seq_id	(0 < Scheduled_Sequence_Id)
	pk_contact_Schedules	Scheduled_Sequence_Id
	nn_cs_sched_start_time	Scheduled_Start_Time
	nn_cs_sched_Stop_Time	Scheduled_Stop_Time
IDP_Acct	nn_ia_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_IDP_Acct	Sub_Intv(Sub_Intv_Sequence_Id)
	ck_ia_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_ia_scene_center_time	Scene_Center_Time
	ck_ia_scene_center_time	(Sub_Intv_Scene_Number

		between 1 and 99)

Table 12–4. LPS Database Table Constraints (4 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_ia_scene_center_scan_num	Scene_Center_Scan_Number
	ck_ia_scene_center_scan_num	(Scene_Center_Scan_Number between -175 and 11725)
	ck_ia_cca_q1_score	(CCA_Quadrant1_Score between 0.00 and 100.00)
	ck_ia_cca_q2_score	(CCA_Quadrant2_Score between 0.00 and 100.00)
	ck_ia_cca_q3_score	(CCA_Quadrant3_Score between 0.00 and 100.00)
	ck_ia_cca_q4_score	(CCA_Quadrant4_Score between 0.00 and 100.00)
	ck_ia_cca_aggr_score	(CCA_Aggregate_Score between 0.00 and 100.00)
	ck_ia_full_partial_scene	(Full_Partial_Scene = F or Full_Partial_Scene = P)
	pk_IDP_Acct	(Sub_Intv_Sequence_Id, Sub_Intv_Scene_Number)
LDT_DAN_Info	nn_ldi_dan_seq_num	DAN_Sequence_Num
	pk_ldt_dan_info	DAN_Sequence_Num
	ck_ldi_dan_seq_num	(0 < DAN_Sequence_Num)
	nn_ldi_cont_seq_id	Contact_Sequence_Id
	ck_ldi_cont_seq_id	(0 < Contact_Sequence_Id)
	nn_ldi_file_ver_num	File_Version_Number
	ck_ldi_file_ver_num	(0 <= File_Version_Number)
	nn_ldi_dan_ack_time_stamp	DAN_Ack_Time_Stamp
	nn_ldi_dan_status	DAN_Status
	ck_ldi_dan_status	(DAN_Status between 0 and 6)
	nn_ldi_num_sub_intv	Num_Sub_Intervals
	ck_ldi_num_sub_intv	(Num_Sub_Intervals between 0 and 99)
	nn_ldi_ddn_status	DDN_Status
	ck_ldi_ddn_status	(DDN_Status between 0 and 3)
	fk_LDT_DAN_Info	processing_version_info (Contact_Sequence_Id, File_Version_Number)
LDT_DAN_Transfer_State	nn_ldts_xfer_state	Xfer_State
	ck_ldts_xfer_state	(Xfer_State between 0 and 1)
LDT_DDN_Server_Info	nn_ldsi_process_name	Process_Name

Table 12–4. LPS Database Table Constraints (5 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_ldsi_process_id	Process_Id
	ck_ldsi_process_id	(0 <= Process_Id)
	nn_ldsi_processing_stage	Processing_Stage
	ck_ldsi_processing_stage	(Processing_Stage between 0 and 5)
LDT_File_Group_Info	nn_lfgi_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_LDT_File_Group_Info	Sub_Intv(Sub_Intv_Sequence_Id)
	pk_ldt_file_group_info	Sub_Intv_Sequence_Id
	ck_lfgi_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_lfgi_num_files	Number_Of_Files
	ck_lfgi_num_files	(0 <= Number_Of_Files)
	nn_lfgi_file_gr_daa_disp	File_Group_DAA_Disposition
	nn_lfgi_file_gr_xfer_status	File_Group_Xfer_Status
	ck_lfgi_file_gr_xfer_status	(File_Group_Xfer_Status between 0 and 1)
	nn_lfgi_file_gr_del_status	File_Group_Deletion_Status
	ck_lfgi_fil_gr_del_status	(File_Group_Deletion_Status between 0 and 2)
	nn_lfgi_num_files_deleted	Number_Of_Files_Deleted
	ck_lfgi_num_files_deleted	(0 <= Number_Of_Files_Deleted)
LDT_File_Set_Info	nn_lfsi_cont_seq_id	Contact_Sequence_Id
	ck_lfsi_cont_seq_id	(0 < Contact_Sequence_Id)
	nn_lfsi_file_ver_num	File_Version_Number
	ck_lfsi_file_ver_num	(0 <= File_Version_Number)
	nn_lfsi_retention_status	Retention_Status
	ck_lfsi_retention_status	(Retention_Status between 0 and 1)
	nn_lfsi_num_file_groups	Num_File_Groups
	ck_lfsi_num_file_groups	(0 <= Num_File_Groups)
	nn_lfsi_num_file_gr_daa_fail	Num_File_Groups_DAA_Failed
	ck_lfsi_num_file_gr_daa_fail	(0 <= Num_File_Groups_DAA_Failed)
	nn_lfsi_num_file_gr_xfer_fail	Num_File_Groups_Xfer_Failed
	ck_lfsi_num_file_gr_xfer_fail	(0 <= Num_File_Groups_Xfer_Failed)
	nn_lfsi_file_set_xfer_status	File_Set_Xfer_Status
	ck_lfsi_file_set_xfer_status	(File_Set_Xfer_Status between

		0 and 4)

Table 12–4. LPS Database Table Constraints (6 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_lfsi_file_set_xfer_disp	File_Set_Xfer_Disposition
	ck_lfsi_file_set_xfer_disp	(0 <= File_Set_Xfer_Disposition)
	nn_lfsi_file_set_daa_status	File_Set_DAA_Status
	ck_lfsi_file_set_daa_status	(File_Set_DAA_Status between 0 and 4)
	nn_lfsi_file_set_del_status	File_Set_Deletion_Status
	ck_lfsi_file_set_del_status	(File_Set_Deletion_Status between 0 and 2)
	nn_lfsi_num_file_gr_del	Num_File_Groups_Deleted
	ck_lfsi_num_file_gr_del	(Num_File_Groups_Deleted >= 0)
	pk_ldt_file_set_info	(Contact_Sequence_Id, File_Version_Number)
	fk_ldt_file_set_info	processing_version_info (Contact_Sequence_Id, File_Version_Number)
LPS_Configuration	nn_lc_lps_hw_str_id	LPS_Hardware_String_Id
	nn_lc_ecs_hw_str_id	ECS_Hardware_String_Id
	nn_lc_capt_src	Capture_Source
	nn_lc_spacecraft_id	Spacecraft_Id
	nn_lc_instrument_id	LPS_Software_Ver_Num
	nn_lc_lps_user_id	LPS_User_Id
	nn_lc_lps_password	LPS_Password
	nn_lc_ecs_user_id	ECS_User_Id
	nn_lc_ecs_password	ECS_Password
	nn_lc_lps_port_num	LPS_Port_Num
	nn_lc_ecs_port_num	ECS_Port_Num
	nn_lc_ias_param_file_ver_num	IAS_Param_File_Ver_Num
LPS_Contact_ Sched_Files	nn_lcsf_file_name	File_Name
LPS_File_Info	nn_lfi_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_LPS_File_Info	Sub_Intv(Sub_Intv_Sequence_Id)
	ck_lfi_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_lfi_file_patb	File_Path
	nn_lfi_file_name	File_Name
	nn_lfi_file_type	File_Type
	nn_lfi_file_del_status	File_Deletion_status
	ck_lfi_file_del_status	(File_Deletion_status between 0

		and 3)

Table 12–4. LPS Database Table Constraints (7 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	ck_lfi_file_xfer_disp	(0 <= File_Xfer_Disposition)
	pk_lps_file_info	(Sub_Intv_Sequence_Id, File_Name)
LPS_IAS_Parm_File	nn_lipf_file_name	File_Name
MFP_Acct	nn_ma_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_MFP_Acct	Sub_Intv(Sub_Intv_Sequence_Id)
	ck_ma_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_ma_mjf_cadu_rcvd_cnt	Mjf_CADU_Rcvd_Cnt
	ck_ma_mjf_cadu_rcvd_cnt	(0 <= Mjf_CADU_Rcvd_Cnt)
	nn_ma_mjf_cadu_fly_cnt	Mjf_CADU_Fly_Cnt
	ck_ma_mjf_cadu_fly_cnt	(0 <= Mjf_CADU_Fly_Cnt)
	nn_ma_mjf_cadu_inv_pol_cnt	Mjf_CADU_Inverted_Polarity_Cnt
	ck_ma_mjf_cadu_inv_pol_cnt	(0 <= Mjf_CADU_Inverted_Polarity_Cnt)
	nn_ma_mjf_cadu_pol_change_cnt	Mjf_CADU_Polarity_Change_Cnt
	ck_ma_mjf_cadu_pol_change_cnt	(0 <= Mjf_CADU_Polarity_Change_Cnt)
	nn_ma_mjf_cadu_bit_slip_cnt	Mjf_CADU_Bit_Slip_Cnt
	ck_ma_mjf_cadu_bit_slip_cnt	(0 <= Mjf_CADU_Bit_Slip_Cnt)
	nn_ma_mjf_cadu_sync_arr_cnt	Mjf_CADU_Sync_Arr_Cnt
	ck_ma_mjf_cadu_sync_arr_cnt	(0 <= Mjf_CADU_Sync_Arr_Cnt)
	nn_ma_mjf_cadu_missing_cnt	Mjf_CADU_Missing_Cnt
	ck_ma_mjf_cadu_missing_cnt	(0 <= Mjf_CADU_Missing_Cnt)
	nn_ma_mjf_cadu_rs_corr_cnt	Mjf_CADU_RS_Corr_Cnt
	ck_ma_mjf_cadu_rs_corr_cnt	(0 <= Mjf_CADU_RS_Corr_Cnt)
	nn_ma_mjf_cadu_rs_uncorr_cnt	Mjf_CADU_RS_Uncorr_Cnt
	ck_ma_mjf_cadu_rs_uncorr_cnt	(0 <= Mjf_CADU_RS_Uncorr_Cnt)
	nn_ma_mjf_bch_data_corr_cn	Mjf_CADU_BCH_Data_Corr_Cnt
	ck_ma_mjf_bch_data_corr_cnt	(0 <= Mjf_CADU_BCH_Data_Corr_Cnt)
	nn_ma_mjf_bch_data_uncorr_cnt	Mjf_CADU_BCH_Data_Uncorr_Cnt
	ck_ma_mjf_bch_data_uncorr_cnt	(0 <= Mjf_CADU_BCH_Data_Uncorr_Cnt)

	nn_ma_mjf_cadu_bits_corr_cnt	Mjf_CADU_Data_Bits_Corr_Cnt
	ck_ma_mjf_cadu_bits_corr_cnt	(0 <= Mjf_CADU_Data_Bits_Corr_Cnt)

Table 12–4. LPS Database Table Constraints (8 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_ma_mjf_bch_ptr_corr_cnt	Mjf_CADU_BCH_Ptr_Corr_Cnt
	ck_ma_mjf_bch_ptr_corr_cnt	(0 <= Mjf_CADU_BCH_Ptr_ Corr_Cnt)
	nn_ma_mjf_bch_ptr_uncorr_cn	Mjf_CADU_BCH_Ptr_Uncorr_Cn t
	ck_ma_mjf_bch_ptr_uncorr_cnt	(0 <= Mjf_CADU_BCH_Ptr_ Uncorr_Cnt)
	nn_ma_mjf_bch_ptr_bits_cr_cn t	Mjf_CADU_BCH_Ptr_Bits_Corr_ Cnt
	ck_ma_mjf_bch_ptr_bits_cr_cn t	(0 <= Mjf_CADU_BCH_Ptr_Bits_ Corr_Cnt)
	nn_ma_mjf_cadu_crc_err_cnt	Mjf_CADU_CRC_Err_Cnt
	ck_ma_mjf_cadu_crc_err_cnt	(0 <= Mjf_CADU_CRC_Err_Cnt)
	nn_ma_mjf_cnt	Mjf_Cnt
	ck_ma_mjf_cnt	(0 <= Mjf_Cnt)
	nn_ma_mjf_time_code_err_cnt	Mjf_Time_Code_Err_Cnt
	ck_ma_mjf_time_code_err_cnt	(0 <= Mjf_Time_Code_Err_Cnt)
	nn_ma_mjf_full_fill_cnt	Mjf_Full_Fill_Cnt
	ck_ma_mjf_full_fill_cnt	(0 <= Mjf_Full_Fill_Cnt)
	nn_ma_mjf_part_fill_cnt	Mjf_Part_Fill_Cnt
	ck_ma_mjf_part_fill_cnt	(0 <= Mjf_Part_Fill_Cnt)
MFP_MJF_Acct	nn_mma_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_MFP_MJF_Acct	Sub_Intv(Sub_Intv_Sequence_I d)
	ck_mma_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_mma_mfp_mjf_time	MFP_MJF_Time
	nn_mma_cadu_received	CADUs_Received
	ck_mma_cadu_received	(CADUs_Received between 0 and 999999)
	nn_mma_fly_wheel_cadus	Fly_Wheel_CADUs
	ck_mma_fly_wheel_cadus	(Fly_Wheel_CADUs between 0 and 999999)
	nn_mma_rs_error_vcdus	RS_Error_VCDUs
	ck_mma_rs_error_vcdus	(RS_Error_VCDUs between 0 and 999999)
	nn_mma_bch_corr_vcdus	BCH_Corr_VCDUs
	ck_mma_bch_corr_vcdus	(BCH_Corr_VCDUs between 0 and 999999)
	nn_mma_bch_uncorr_vcdus	BCH_Uncorr_VCDUs
	ck_mma_bch_uncorr_vcdus	(BCH_Uncorr_VCDUs between 0

		and 999999)

Table 12–4. LPS Database Table Constraints (9 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_mma_num_bit_errors	Num_Bit_Errors
	ck_mma_num_bit_errors	(Num_Bit_Errors between 0 and 999999)
	nn_mma_etm_timecode_error_flag	ETM_Timecode_Error_Flag
	ck_mma_etm_timecode_error_flag	(ETM_Timecode_Error_Flag between 0 and 1)
	nn_mma_mjf_filled_flag	MJF_Filled_Flag
	ck_mma_mjf_filled_flag	(MJF_Filled_Flag = E, P, or N)
	pk_MFP_MJF_Acct	(Sub_Intv_Sequence_Id, MFP_MJF_Time)
PCD_Acct	nn_pa_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_PCD_Acct	Sub_Intv (Sub_Intv_Sequence_Id)
	ck_pa_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	pk_PCD_Acct	Sub_Intv_Sequence_Id
	nn_pa_num_pcd_mjf	Num_PCD_MJF
	ck_pa_num_pcd_mjf	(0 <= Num_PCD_MJF)
	nn_pa_first_pcd_mjf_time	First_PCD_MJF_Time
	nn_pa_pcd_stop_time	PCD_Stop_Time
	nn_pa_etm_last_on_time	ETM_Last_On_Time
	nn_pa_etm_last_off_time	ETM_Last_Off_Time
	nn_pa_ut1_corrections	UT1_Corrections
	ck_pa_ut1_corrections	(UT1_Corrections between -0.90000 and 0.90000)
PCD_MJF_Acct	nn_pma_sub_intv_seq_id	Sub_Intv_Sequence_Id
	fk_PCD_MJF_Acct	Sub_Intv (Sub_Intv_Sequence_Id)
	ck_pma_sub_intv_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_pma_pcd_mjf_time	PCD_MJF_Time
	nn_pma_pcd_words_received	PCD_Words_Receive
	ck_pma_pcd_words_received	(PCD_Words_Received between 0 and 999999)
	nn_pma_total_pcd_mnf	Total_PCD_MNF
	ck_pma_total_pcd_mnf	(Total_PCD_MNF between 0 and 999)
	nn_pma_failed_pcd_votes	Failed_PCD_Votes
	ck_pma_failed_pcd_votes	(Failed_PCD_Votes between 0 and 999999)

Table 12–4. LPS Database Table Constraints (10 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_pma_num_pcd_mnf_sync_err	Num_PCD_MNF_Sync_Errors
	ck_pma_num_pcd_mnf_sync_err	(Num_PCD_MNF_Sync_Errors between 0 and 999)
	nn_pma_num_pcd_filled_mnf	Num_PCD_Filled_MNF
	ck_pma_num_pcd_filled_mnf	(Num_PCD_Filled_MNF between 0 and 999)
	nn_pma_pcd_filled_mjf_flag	PCD_Filled_MJF_Flag
	ck_pma_pcd_filled_mjf_flag	(PCD_Filled_MJF_Flag = E or P
	nn_pma_pcd_adp_mjf_flag	PCD_ADP_MJF_Flag
	ck_pma_pcd_adp_mjf_flag	(PCD_ADP_MJF_Flag = G, R, or M)
	nn_pma_pcd_edp_mjf_flag	PCD_EDP_MJF_Flag
	ck_pma_pcd_edp_mjf_flag	(PCD_EDP_MJF_Flag = G, R, or M)
	pk_pcd_mjf_acct	(Sub_Intv_Sequence_Id, PCD_MJF_Time)
PCD_Scene_Acct	nn_psa_sub_int_seq_id	Sub_Intv_Sequence_Id
	fk_PCD_Scene_Acct	Sub_Intv (Sub_Intv_Sequence_Id)
	ck_psa_sub_int_seq_id	(0 < Sub_Intv_Sequence_Id)
	nn_psa_scene_center_time	Scene_Center_Time
	nn_psa_wrs_path_nom	WRS_Path_Nominal
	ck_psa_wrs_path_nom	(WRS_Path_Nominal between 0 and 233)
	nn_psa_wrs_row_nom	WRS_Row_Nominal
	ck_psa_wrs_row_nom	(WRS_Row_Nominal between 0 and 248)
	nn_psa_hori_disp_shift	Horizontal_Display_Shift
	ck_psa_hori_disp_shift	(Horizontal_Display_Shift between -9999.0000 and 9999.0000)
	nn_psa_sun_azimuth	Sun_Azimuth
	ck_psa_sun_azimuth	(Sun_Azimuth between -180.0000000 and 180.0000000)
	nn_psa_sun_elev	Sun_Elevation
	ck_psa_sun_elev	(Sun_Elevation between -9999.0000 and 9999.0000)
	nn_psa_cal_door_act_status	Cal_Door_Activity_Status
	ck_psa_cal_door_act_status	(Cal_Door_Activity_Status

		between 0 and 1)

Table 12–4. LPS Database Table Constraints (11 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_psa_scene_cen_lat	Scene_Center_Lat
	ck_psa_scene_cen_lat	(Scene_Center_Lat between -90.0000 and 90.0000)
	nn_psa_scene_cen_lon	Scene_Center_Lon
	ck_psa_scene_cen_lon	(Scene_Center_Lon between -180.0000 and 180.0000)
	nn_psa_scene_ul_lat	Scene_Upper_Left_Lat
	ck_psa_scene_ul_lat	(Scene_Upper_Left_Lat between -90.0000 and 90.0000)
	nn_psa_scene_ul_lo	Scene_Upper_Left_Lon
	ck_psa_scene_ul_lon	(Scene_Upper_Left_Lon between -180.0000 and 180.0000)
	nn_psa_scene_ur_lat	Scene_Upper_Right_Lat
	ck_psa_scene_ur_lat	(Scene_Upper_Right_Lat between -90.0000 and 90.0000
	nn_psa_scene_ur_lon	Scene_Upper_Right_Lon
	ck_psa_scene_ur_lon	(Scene_Upper_Right_Lon between -180.0000 and 180.0000
	nn_psa_scene_ll_lat	Scene_Lower_Left_Lat
	ck_psa_scene_ll_lat	(Scene_Lower_Left_Lat between -90.0000 and 90.0000)
	nn_psa_scene_ll_lon	Scene_Lower_Left_Lon
	ck_psa_scene_ll_lon	(Scene_Lower_Left_Lon between -180.0000 and 180.0000)
	nn_psa_scene_lr_lat	Scene_Lower_Right_Lat
	ck_psa_scene_lr_lat	(Scene_Lower_Right_Lat between -90.0000 and 90.0000)
	nn_psa_scene_lr_lon	Scene_Lower_Right_Lon
	ck_psa_scene_lr_lon	(Scene_Lower_Right_Lon between -180.0000 and 180.0000)
	pk_PCD_Scene_Acct	(Sub_Intv_Sequence_Id, Scene_Center_Time)
Process_Id	ck_pi_cont_seq_id	(0 < Contact_Sequence_Id)
	ck_pi_file_ver_num	(0 <= File_Version_Number)
	nn_pi_par_proc_id	Parent_Process_Id
	pk_process_id	Parent_Process_Id
	nn_pi_task_name	Task_Name

RDP_Acct	nn_rdpa_cont_seq_id	Contact_Sequence_Id
	ck_rdpa_contact_seq_id	(0 < Contact_Sequence_Id)

Table 12–4. LPS Database Table Constraints (12 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_rdpa_file_ver_num	File_Version_Number
	ck_rdpa_file_ver_num check	(0 <= File_Version_Number)
	nn_rdpa_ccsds_parameter_id	CCSDS_Parms_Id
	ck_rdpa_ccsds_parameter_id	(0 < CCSDS_Parms_Id)
	nn_rdpa_inv_cadu_cnt	Inverted_CADU_Cnt
	ck_rdpa_inv_cadu_cnt	(0 <= Inverted_CADU_Cnt)
	nn_rdpa_pol_change_cnt	Polarity_Change_Cnt
	ck_rdpa_pol_change_cnt	(0 <= Polarity_Change_Cnt)
	nn_rdpa_cadu_bit_slip_cnt	CADU_Bit_Slip_Cnt
	ck_rdpa_cadu_bit_slip_cnt	(0 <= CADU_Bit_Slip_Cnt)
	nn_rdpa_cadu_sync_err_cnt	CADU_Sync_Err_Cnt
	ck_rdpa_cadu_sync_err_cnt	(0 <= CADU_Sync_Err_Cnt)
	nn_rdpa_cadu_rcv_cnt	CADU_Rcv_Cnt
	ck_rdpa_cadu_rcv_cnt	(0 <= CADU_Rcv_Cnt)
	nn_rdpa_cadu_fw_cnt	CADU_Flywheel_Cnt
	ck_rdpa_cadu_fw_cnt	(0 <= CADU_Flywheel_Cnt)
	nn_rdpa_fill_cadu_cnt	Fill_CADU_Cnt
	ck_rdpa_fill_cadu_cn	(0 <= Fill_CADU_Cnt)
	nn_rdpa_cadu_crc_err_cnt	CADU_CRC_Err_Cnt
	ck_rdpa_cadu_crc_err_cnt	(0 <= CADU_CRC_Err_Cnt)
	nn_rdpa_vcdu_h1_corr_err_cnt	VCDU_Header1_Corr_Err_Cnt
	ck_rdpa_vcdu_h1_corr_err_cnt	(0 <= VCDU_Header1_Corr_Err_Cnt)
	nn_rdpa_vcdu_h2_corr_err_cnt	VCDU_Header2_Corr_Err_Cnt
	ck_rdpa_vcdu_h2_corr_err_cnt	(0 <= VCDU_Header2_Corr_Err_Cnt)
	nn_rdpa_vcdu_hd_uncorr_err_cnt	VCDU_Header_Uncorr_Err_Cnt
	ck_rdpa_vcdu_hd_uncorr_err_cnt	(0 <= VCDU_Header_Uncorr_Err_Cnt)
	nn_rdpa_bch_data_corr_err_cnt	BCH_Data_Corrected_Err_Cnt
	ck_rdpa_bch_data_corr_err_cnt	(0 <= BCH_Data_Corrected_Err_Cnt)
	nn_rdpa_bch_data_ucr_err_cnt	BCH_Data_Uncorrected_Err_Cnt
	ck_rdpa_bch_data_ucr_err_cnt	(0 <= BCH_Data_Uncorrected_Err_Cnt)
	nn_rdpa_bch_data_corr_bit_cnt	BCH_Data_Corrected_Bits_Cnt

	ck_rdpa_bch_data_corr_bit_cnt	(0 <= BCH_Data_Corrected_Bits_Cnt)

Table 12–4. LPS Database Table Constraints (13 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_rdpa_bch_ptr_corr_err_cnt	BCH_Ptr_Corrected_Err_Cnt
	ck_rdpa_bch_ptr_corr_err_cnt	(0 <= BCH_Ptr_Corrected_ Err_Cnt)
	nn_rdpa_bch_ptr_uncorr_err_cn	BCH_Ptr_Uncorrected_Err_Cnt
	ck_rdpa_bch_ptr_uncorr_err_cnt	(0 <= BCH_Ptr_Uncorrected_ Err_Cnt)
	nn_rdpa_bch_ptr_corr_bit_cnt	BCH_Ptr_Corrected_Bit_Cnt
	ck_rdpa_bch_ptr_corr_bit_cnt	(0 <= BCH_Ptr_Corrected_ Bit_Cnt
	pk_RDP_Acct	(Contact_Sequence_Id, File_Version_Number)
	fk_RDP_Acct	processing_version_info (Contact_Sequence_Id, File_Version_Number)
UTC_UT1_ Corrections	nn_uuc_corr_date	Correction_date
	pk_UTC_UT1_Corrections	Correction_date
	nn_uuc_ut1_corr	UT1_Corrections
	ck_uuc_ut1_corr	(UT1_Corrections between -0.90000 and 0.90000)
Valid_Band_Parms	nn_vbp_multi1	Multi1
	ck_vbp_multi1	(multi1 between 0 and 6)
	nn_vbp_multi2	Multi2
	ck_vbp_multi2	(multi2 between 0 and 6)
	nn_vbp_multi3	Multi3
	ck_vbp_multi3	(multi3 between 0 and 6)
	nn_vbp_subs	Subs
	ck_vbp_subs	(Subs >= 0)
	nn_vbp_wave	Wave
	ck_vbp_wave	(Wave >= 0)
	nn_vbp_jpeg_qlty	JPEG_Quality
	ck_vbp_jpeg_qlty	(JPEG_Quality between 0 and 100)
	nn_vbp_cont_str_fact	Contrast_Stretch_Factor
	ck_vbp_cont_str_fact	(Contrast_Stretch_Factor between 0 and 49)
Valid_CCSDS_ Parms	nn_vcp_ccsds_parms_id	CCSDS_Parms_Id
	pk_valid_ccsds_parms	CCSDS_Parms_Id
	nn_vcp_insertion_time	Insertion_Time

Table 12–4. LPS Database Table Constraints (14 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_vcp_cadu_search_tol	CADU_Search_Tol
	ck_vcp_cadu_search_tol	(0 <= CADU_Search_Tol and CADU_Search_Tol <= 3)
	nn_vcp_cadu_check_tol	CADU_Check_Tol
	ck_vcp_cadu_check_tol	(0 <= CADU_Check_Tol and CADU_Check_Tol < = 3)
	nn_vcp_cadu_fw_tol	CADU_Flywheel_Tol
	ck_vcp_cadu_fw_tol	(CADU_Flywheel_Tol between 0 and 3)
	nn_vcp_cadu_sync_ck_err_tol	CADU_Sync_Mark_Check_Error _Tol
	ck_vcp_cadu_sync_ck_err_tol	(CADU_Sync_Mark_Check_ Error_Tol between 0 and 3)
	nn_vcp_cadu_sync_lock_err_to l	CADU_Sync_Lock_Error_Tol
	ck_vcp_cadu_sync_lock_err_tol	(CADU_Sync_Lock_Error_Tol between 0 and 3)
	nn_vcp_cadu_bit_slip_corr_ext	CADU_Bit_Slip_Corr_Extent
	ck_vcp_cadu_bit_slip_corr_ext	(CADU_Bit_Slip_Corr_Extent between -3 and 3)
	nn_vcp_xfer_fr_tro_file_num	Xfer_Frame_Trouble_File
	ck_vcp_xfer_fr_tro_file_num	(Xfer_Frame_Trouble_File between 0 and 1)
	nn_vcp_bch_flag	BCH_Flag number
	ck_vcp_bch_flag	(BCH_Flag between 0 and 1)
Valid_Detector_ Gain_Bias	nn_vdgb_band	Band
	ck_vdgb_band	(Band between 1 and 6)
	nn_vdgb_detector	Detector
	ck_vdgb_detector	(Detector between 1 and 16)
	nn_vdgb_high_gain	High_Gain
	nn_vdgb_high_gain_bias	High_Gain_Bias
	nn_vdgb_low_gain	Low_Gain
	nn_vdgb_low_gain_bias	Low_Gain_Bias
	pk_Valid_Detector_Gain_Bias	(Band, Detector)
Valid_LDT_Parms	nn_vlp_num_auth_req	Num_Auth_Request
	ck_vlp_num_auth_request	(0 <= Num_Auth_Request)
	nn_vlp_timeout_auth_req	Timeout_Auth_Request
	ck_vlp_timeout_auth_req	(0 <= Timeout_Auth_Request
	nn_vlp_nsda_num_send_dan_a	Num_Send_DAN_Attempt

	tt	

Table 12–4. LPS Database Table Constraints (15 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	ck_vlp_nsda_num_send_dan_att	(0 <= Num_Send_DAN_Attempt),
	nn_vlp_timeout_send_dan_att	Timeout_Send_DAN_Attempt
	ck_vlp_timeout_send_dan_at	(0 <= Timeout_Send_DAN_Attempt)
	nn_vlp_timeout_recv_daa	Timeout_Receive_DAA
	ck_vlp_timeout_recv_daa	(0 <= Timeout_Receive_DAA)
	nn_vlp_timeout_recv_ddn	Timeout_Receive_DDN
	ck_vlp_timeout_recv_ddn	(0 <= Timeout_Receive_DDN)
	nn_vlp_num_send_dda_att	Num_Send_DDA_Attempt
	ck_vlp_num_send_dda_att	(0 <= Num_Send_DDA_Attempt)
	nn_vlp_timeout_send_dda	Timeout_Send_DDA
	ck_vlp_timeout_send_dda	(0 <= Timeout_Send_DDA)
	nn_vlp_read_sleep_second	Read_Sleep_Second
	ck_vlp_read_sleep_second	(0 <= Read_Sleep_Second)
	nn_vlp_ldt_ecs_msg_save	LDT_ECS_Msg_Save
	ck_vlp_ldt_ecs_msg_save	(LDT_ECS_Msg_Save between 0 and 1)
Valid_MFP_Parms	nn_vmp_fill_val	Fill_Value
	nn_vmp_sub_intv_delta	Sub_Intv_Delta
	ck_vmp_sub_intv_delta	(Sub_Intv_Delta > 0)
	nn_vmp_mjf_data_period	Mjf_Data_Period
	ck_vmp_mjf_data_period	(Mjf_Data_Period > 0)
	nn_vmp_time_range_period	Time_Range_Tol
	ck_vmp_time_range_period	(Time_Range_Tol > 0)
	nn_vmp_maj_vote_tol	Maj_Vote_Tol
	ck_vmp_maj_vote_tol	(Maj_Vote_Tol >= 0)
	nn_vmp_max_time_span	Max_Time_Span
	ck_vmp_max_time_span	(Max_Time_Span >= 0)
	nn_vmp_eol_tol	Eol_Tol
	ck_vmp_eol_tol	(Eol_Tol >= 0)
	nn_vmp_mjf_sync_tol	Mjf_Sync_Tol
	ck_vmp_mjf_sync_tol	(Mjf_Sync_Tol >= 0)
	nn_vmp_mjf_sync_size	Mjf_Sync_Size
	ck_vmp_mjf_sync_size	(Mjf_Sync_Size >= 0)
	nn_vmp_max_mnf_counter	Max_Mnf_Counter
	ck_vmp_max_mnf_counter	(Max_Mnf_Counter >= 0)

	nn_vmp_etm_plus_trob_file	ETM_Plus_Trouble_File

Table 12–4. LPS Database Table Constraints (16 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	ck_vmp_etm_plus_trob_file	(ETM_Plus_Trouble_File between 0 and 1)
	nn_vmp_min_mjf_cnt_sub_intv	Min_Mjf_Count_Per_Sub_Intv
	ck_vmp_min_mjf_cnt_sub_intv	(Min_Mjf_Count_Per_Sub_Intv > 0)
Valid_MFP_Thres	nn_vmt_mjf_cadu_seq_err_thre s	MJF_CADU_Seq_Err_Thres
	ck_vmt_mjf_cadu_seq_err_thre s	(MJF_CADU_Seq_Err_Thres >= 0)
	nn_vmt_mjf_sync_thres	Mjf_Sync_Thres
	ck_vmt_mjf_sync_thres	(Mjf_Sync_Thres >= 0)
	nn_vmt_mnf_ctr_thres	Mnf_Ctr_Thres
	ck_vmt_mnf_ctr_thres	(Mnf_Ctr_Thres >= 0)
	nn_vmt_eol_thres	Eol_Thres
	ck_vmt_eol_thres	(Eol_Thres >= 0)
	nn_vmt_tc_thres	Tc_Thres
	ck_vmt_tc_thres	(Tc_Thres >= 0)
	nn_vmt_full_mjf_thres	Full_Mjf_Thres
	ck_vmt_full_mjf_thres	(Full_Mjf_Thres >= 0)
	nn_vmt_part_mjf_thres	Part_Mjf_Thres
	ck_vmt_part_mjf_thres	(Part_Mjf_Thres >= 0)
	nn_vmt_sub_intv_size_thres	Sub_Intv_Size_Thres
	ck_vmt_sub_intv_size_thres	(Sub_Intv_Size_Thres >= 0)
Valid_MWD_Parms	nn_vmdp_fmt1_red_band	Fmt1_Red_Band
	nn_vmdp_fmt1_green_band	Fmt1_Green_Band
	nn_vmdp_fmt1_blue_band	Fmt1_Blue_Band
	nn_vmdp_fmt2_red_band	Fmt2_Red_Band
	nn_vmdp_fmt2_green_band	Fmt2_Green_Band
	nn_vmdp_fmt2_blue_band	Fmt2_Blue_Band
	nn_vmdp_xoptions	Xoptions
Valid_PCD_Parms	nn_vpp_pcd_frame_fill	PCD_Frame_Fill
	ck_vpp_pcd_frame_fill	(PCD_Frame_Fill >= 0)
Valid_PCD_Thres	nn_vpt_ep_pos_upper	Ephem_Position_Upper
	ck_vpt_ep_pos_upper	(Ephem_Position_Upper between -83886.6 and 8388.6)
	nn_vpt_ep_pos_lower	Ephem_Position_Lower
	ck_vpt_ep_pos_lower	(Ephem_Position_Lower between -83886.6 and 8388.6)
	nn_vpt_ep_vel_upper	Ephem_Velocity_Upper

Table 12–4. LPS Database Table Constraints (17 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	ck_vpt_ep_vel_upper	(Ephem_Velocity_Upper between -8 and 8)
	nn_vpt_ep_vel_lower	Ephem_Velocity_Lower
	ck_vpt_ep_vel_lower	(Ephem_Velocity_Lower between -8 and 8)
	nn_vpt_ep_xp_max	Ephem_Crossproduct_Max
	ck_vpt_ep_xp_max	(Ephem_Crossproduct_Max between 53000 and 53200)
	nn_vpt_ep_xp_min	Ephem_Crossproduct_Min
	ck_vpt_ep_xp_min	(Ephem_Crossproduct_Min between 53000 and 53200)
	nn_vpt_att_qt_tol	Attitude_Quaternion_Tol
	ck_vpt_att_qt_tol	(Attitude_Quaternion_Tol >= 0)
	nn_vpt_num_miss_data_words	Num_Missing_Data_Words
	ck_vpt_num_miss_data_words	(Num_Missing_Data_Words >= 0)
	nn_vpt_num_failed_votes	Num_Failed_Votes
	ck_vpt_num_failed_votes	(Num_Failed_Votes >= 0)
Valid_RDP_Thres	nn_vrt_Sync_Thres	Sync_Thres
	ck_vrt_Sync_Thres	(0 < Sync_Thres)
	nn_vrt_CRC_Thres	CRC_Thres
	ck_vrt_CRC_Thres	(0 < CRC_Thres)
	nn_vrt_RS_Thres	RS_Thres
	ck_vrt_RS_Thres	(0 < RS_Thres)
	nn_vrt_BCH_Thres	BCH_Thres
	ck_vrt_BCH_Thres	(0 < BCH_Thres)
Valid_Scene_Parms	nn_vsp_ETM_Body_xmtx_11	ETM_Body_Trans_Matrix_11
	ck_vsp_ETM_Body_xmtx_11	(ETM_Body_Trans_Matrix_11 between -1 and 1)
	nn_vsp_ETM_Body_xmtx_12	ETM_Body_Trans_Matrix_12
	ck_vsp_ETM_Body_xmtx_12	(ETM_Body_Trans_Matrix_12 between -1 and 1)
	nn_vsp_ETM_Body_xmtx_13	ETM_Body_Trans_Matrix_13
	ck_vsp_ETM_Body_xmtx_13	(ETM_Body_Trans_Matrix_13 between -1 and 1)
	nn_vsp_ETM_Body_xmtx_21	ETM_Body_Trans_Matrix_21
	ck_vsp_ETM_Body_xmtx_21	(ETM_Body_Trans_Matrix_21 between -1 and 1)
	nn_vsp_ETM_Body_xmtx_22	ETM_Body_Trans_Matrix_22

Table 12–4. LPS Database Table Constraints (18 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	ck_vsp_ETM_Body_xmtx_22	(ETM_Body_Trans_Matrix_22 between -1 and 1)
	nn_vsp_ETM_Body_xmtx_23	ETM_Body_Trans_Matrix_23
	ck_vsp_ETM_Body_xmtx_23	(ETM_Body_Trans_Matrix_23 between -1 and 1)
	nn_vsp_ETM_Body_xmtx_31	ETM_Body_Trans_Matrix_31
	ck_vsp_ETM_Body_xmtx_31	(ETM_Body_Trans_Matrix_31 between -1 and 1)
	nn_vsp_ETM_Body_xmtx_32	ETM_Body_Trans_Matrix_32
	ck_vsp_ETM_Body_xmtx_32	(ETM_Body_Trans_Matrix_32 between -1 and 1)
	nn_vsp_ETM_Body_xmtx_33	ETM_Body_Trans_Matrix_33
	ck_vsp_ETM_Body_xmtx_33	(ETM_Body_Trans_Matrix_33 between -1 and 1)
	nn_vsp_semi_major_axis	Semi_Major_Axis
	ck_vsp_semi_major_axis	(Semi_Major_Axis >= 0)
	nn_vsp_semi_minor_axis	Semi_Minor_Axis
	ck_vsp_semi_minor_axis	(Semi_Minor_Axis >= 0)
Valid_Sensor_Align _Parms	pk_Valid_Sensor_Align_Parms	Sensor_Align_Title
	nn_vsap_sensor_align_title	Sensor_Align_Title
	nn_vsap_sensor_align_value	Sensor_Align_Value
Valid_WRS_Parms	nn_vwp_wrs_path_nominal	WRS_Path_Nominal
	ck_vwp_WRS_Path_Nominal	(WRS_Path_Nominal between 1 and 233)
	nn_vwp_wrs_row_nominal	WRS_Row_Nominal
	ck_vwp_WRS_Row_Nominal	(WRS_Row_Nominal between 1 and 248)
	nn_vwp_SC_Lat	Scene_Center_Latitude
	ck_vwp_SC_Lat	(Scene_Center_Latitude between -90.00 and 90.00)
	nn_vwp_SC_Lon	Scene_Center_Longitude
	ck_vwp_SC_Lon	(Scene_Center_Longitude between -180.00 and 180.00)
	nn_vwp_TLC_Lat	Trailing_Left_Latitude
	ck_vwp_TLC_Lat	(Trailing_Left_Latitude between -90.00 and 90.00)
	nn_vwp_TLC_Lon	Trailing_Left_Longitude
	ck_vwp_TLC_Lon	(Trailing_Left_Longitude between -180.00 and 180.00)

Table 12–4. LPS Database Table Constraints (19 of 19)

Table Name	Constraint Name	Column Name, Check Range, and Foreign Key References
	nn_vwp_TRC_Lat	Trailing_Right_Latitude
	ck_vwp_TRC_Lat	(Trailing_Right_Latitude between -90.00 and 90.00)
	nn_vwp_TRC_Lon	Trailing_Right_Longitude
	ck_vwp_TRC_Lon	(Trailing_Right_Longitude between -180.00 and 180.00)
	nn_vwp_LLC_Lat	Leading_Left_Latitude
	ck_vwp_LLC_La	(Leading_Left_Latitude between -90.00 and 90.00)
	nn_vwp_LLC_Lon	Leading_Left_Longitude
	ck_vwp_LLC_Lon	(Leading_Left_Longitude between -180.00 and 180.00)
	nn_vwp_LRC_Lat	Leading_Right_Latitude
	ck_vwp_LRC_Lat	(Leading_Right_Latitude between -90.00 and 90.00)
	nn_vwp_LRC_Lon	Leading_Right_Longitude
	ck_vwp_LRC_Lon	(Leading_Right_Longitude between -180.00 and 180.00)
	pk_valid_wrs_parms	(WRS_Path_Nominal, WRS_Row_Nominal)

12.5 Database Administration

This section describes database administration procedures to facilitate data security, integrity, backup, and recovery.

12.5.1 Security

LPS database security will be accomplished at two levels: system and data. ORACLE-provided security features will be used to control LPS database accessibility and usage. System security includes the mechanisms that control the access and use of the database at the system level. LPS database system security options include

- Valid user name/password combination
- Authorization to connect to LPS database
- Amount of disk space available to the objects of a user/process
- Resource limits for a user/process
- Audit of designated processes
- Which system operations a user group can perform

Data security includes the mechanisms that control the access and use of the database at the object level. LPS database data security options include

- Which users have access to a specific schema object and the specific types of actions allowed for each user on the object (e.g., user A can issue SELECT and INSERT statements, but not DELETE statements, using the B table)
- Actions, if any, that are audited for each schema level

This data security option information is stored in the LPS data dictionary.

12.5.2 Integrity

The LPS database makes sure that its data adheres to a predefined set of rules as determined by the database administrator or application developer. LPS data integrity is maintained using two types of checking mechanisms:

1. Data integrity between the database and the system
2. Data integrity within the database

There will be two executables for checking data integrity between the database and the system. One will check raw data files and the other will check Level 0R output files. Existing files and their information stored in the database will be checked both ways, from the database to the system and from the system to the database. Each file existing in the system should have its information stored in the database. For information about files stored in the database, its existence in the system will be verified. These executables will run by the operator during the maintenance cycle.

To check data integrity within the database, the declarative approach by defining various integrity constraints (NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, and CHECK) will be used. These integrity constraints are supported by ORACLE.

12.5.3 Backup

ORACLE provides the capability to perform both full and partial backups of a database. A full backup is defined as including all tables and files associated with the LPS database. Because the database must be closed and offline to perform a full backup, this backup should be a scheduled activity. A partial backup, sometime referred as an incremental backup, can occur while the database is online and active. Only specified data is processed during the backup. This capacity permits the storage of selected database tables, without impacting current operations. This procedure should enhance, but not replace, scheduled full backups.

12.5.4 Recovery

Recovery after a system crash (e.g., power failure) is automatically handled by ORACLE as part of the normal database startup procedure. All transactions committed before the system crash are recovered and uncommitted transactions are rolled backed.

In the event of a media failure (e.g., a disk crash or corruption), the data on that device can be restored from a backup copy. The data will be restored to the time of the backup (ORACLE’s Export utility can be used); data modifications that were applied after the backup will be reapplied if the mirrored log files are available (i.e., stored to a different disk).

Section 13. User Interface

13.1 Introduction

The LPS user interface is the primary interface between the LPS subsystems and the LPS operator. The interface uses Oracle SQL forms and UNIX shell commands and runs on X Window-based devices to provide a menu-driven window environment for the operation and management of LPS.

13.1.1 Design Considerations

Due to LPS budget constraints, the user interface will be developed with limited capabilities based on the following assumptions:

- No automatic network interfaces between LPS and LGS, MOC, or IAS for schedules and parameters input
- No extra security provided beyond the build-in securities in UNIX shell and Oracle
- Only one user type available, i.e. operator
- No elaborate shell program provided to buffer the operator from UNIX
- No long-term or trend report provided

13.1.2 Tools

Oracle Cooperative Development Environment (CDE) tools are used for designing the user interface. CDE tools are built on top of a layer called the Oracle Toolkit. This toolkit makes it possible to create applications that run against multiple user interfaces, such as Motif or Windows, while retaining the full native look and feel of the interface. Oracle Menu, Oracle Forms, and Oracle Reports of CDE tools are used to build the LPS user interface.

13.1.2.1 Oracle Menu

Oracle Menu is a forms-based application development tool that provides a single menu interface for running multiple data processing tools.

13.1.2.2 Oracle Forms

Oracle Forms is a forms-based application development tool for quickly building interactive applications that access Oracle 7 data.

13.1.2.3 Oracle Reports

Oracle Reports is an application development tool for building and generating reports that use Oracle 7 data. Oracle Reports is specifically designed for application development.

13.2 User Tasks

This section identifies and briefly describes the primary tasks supported by the user interface.

13.2.1 Contact Scheduling Support

The contact schedules are manually entered into the database using the Contact Schedule form. This form allows the operator to view all the schedules, insert new schedules, and update and delete existing schedules. All of the contact schedules are manually received from the LGS.

13.2.2 Parameters and Thresholds Configuration

The operator receives a file containing IAS parameters from the IAS. The operator uses the Data Processing Parameters form to insert the new IAS parameters or update existing IAS parameters in the database. Level 0R thresholds can be entered and adjusted using the Data Processing Thresholds form. The LPS string configuration parameters can be entered and updated using the LPS String Configuration form.

13.2.3 Test System Functions and External Interfaces

The raw data capture task is tested by sending a raw data file to the capture device. The Test System Functions form allows the operator to select an online raw data file and test the raw data capture device by writing the file to the device.

13.2.4 Startup and Shutdown LPS Tasks

The data capture and data processing tasks can be manually started up and shutdown using forms. The Start Data Capture and Stop Data Capture forms allow operator to start and stop data capture. The Start Data Processing and Stop Data Processing forms allow operator to start and stop data processing.

13.2.5 LPS System Monitoring

LPS tasks send operation messages to the LPS Journal file. The operator uses the Display Operation Message form to create a new X Window and display the latest operation messages. The operator can specify the severity level of the operation messages to be displayed on the X Window.

13.2.6 File Management

Transfer of output files to EDC DAAC can be enabled or disabled from the Enable/Disable DAN Transfer form. This form sets a flag in the database to indicate whether or not to automatically transfer LPS output files. LPS output files can be deleted automatically or retained after files have been transferred to EDC DAAC. The Management Files form allows the operator to set the delete/retain flag in the database. For output files that failed the automatic file transfer, the operator can use the Resend DAN form to manually resend DANs for LPS files.

13.2.7 Report Generation

Data receive summary, data processing, Q&A, and file transfer summary reports can be generated and displayed on an X Window or output to a file to be printed.

13.3 User Interface Architecture

The user interface for the LPS is a graphics-based application to operate the LPS and monitor its status. This interface is developed using Oracle Forms. All functions available in the user interface are invoked by selecting menu items. All logic for retrieving information and controlling the user's input is incorporated in Oracle Forms event triggers. 3GL routines, such as C subroutines (user exits), can be invoked from the triggers to perform file I/O and system service calls. The user controls events by invoking the Oracle Forms functions.

13.3.1 Menus and Control

When the user interface is initiated, the main window is displayed. This window contains a series of pulldown menus that allow the user to invoke various functions of the interface. The menus are discussed in the following subsections. Menu items are selected (invoked) with the mouse by placing the cursor on the name of the desired menu, pressing the left mouse button, moving the cursor to the desired item (with the button still down), and releasing the button.

13.3.1.1 SHUTDOWN Menu Item

When this menu item is selected, all the user interface windows are deleted from the display.

13.3.1.2 SETUP Menu Item

Several submenus are available under SETUP:

- **LPS String Configuration...** activates an Oracle form that allows users to insert, query, and update LPS configuration parameters.
- **Data Processing Thresholds...** activates an Oracle form that allows users to insert, query, and update raw data processing, major frame processing, and PCD processing thresholds.
- **Data Processing Parameters...** activates an Oracle form that allows users to insert, query, and update raw data processing, major frame processing, PCD processing, and image data processing parameters.
- **Contact Schedules...** activates an Oracle form that allows users to insert, query, and update the contact schedules information.

13.3.1.3 TEST Menu Item

The **Send Data...** submenu in TEST activates an Oracle form requesting input parameters for sending data. When the operator selects the OK button on the form, the Oracle form invokes the test data task.

13.3.1.4 CONTROL Menu Item

Several submenus are available under CONTROL:

- **Start Data Capture...** activates an Oracle form requesting input parameters for raw data capture. When the operator selects the OK button on the form, the Oracle form invokes rdc_Main.

- **Stop Data Capture...** activates an Oracle form requesting input parameters for stopping raw data capture. When the operator selects the OK button on the form, the form invokes rdc_ShutDownRDC.

- **Start Copy to Tape...** activates an Oracle form requesting input parameters for the copy-to-tape command. When the operator selects the OK button on the form, the form invokes rdc_Save.

- **Stop Copy to Tape...** activates an Oracle form requesting input parameters to stop coping to tape. When the operator selects the OK button on the form, the form invokes rdc_StopSaveRestage.

- **Start Copy from Tape...** activates an Oracle form requesting input parameters for the copy-from-tape command. When the operator selects the OK button on the form, the form invokes rdc_RestageCptr.

- **Stop Copy from Tape...** activates an Oracle form requesting input parameters to stop copying from tape. When the operator selects the OK button on the form, the form invokes rdc_StopSaveRestage.

- **Start Data Processing...** activates an Oracle form requesting input parameters for data processing. When the operator selects the OK button on the form, the form invokes mac_MainStartL0R.

- **Stop Data Processing...** activates an Oracle form requesting input parameters to stop data processing. When the operator selects the OK button on the form, the form invokes mac_MainStopL0R.

13.3.1.5 MONITOR Menu Item

Two submenus are available under MONITOR:

- **Display LPS Journal File...** activates an Oracle form requesting the file range of the Journal file to be displayed. When the operator selects the OK button on the form, the form brings up an X Window and displays the Journal file.

- **Display Operation Messages...** activates an Oracle form requesting the severity level of the operation messages to be displayed. When the operator selects the OK button on the form, the form brings up an X Window and displays the operation messages.

13.3.1.6 FILES Menu Item

Three submenus are available under FILES:

- **Enable/Disable DAN Transfer...** activates an Oracle form requesting enabling or disabling the DAN transfer. When the operator selects the OK button on the form, the form updates the transfer_state field in the LDT_DAN_Transfer_State table.

- **Resend DAN...** activates an Oracle form requesting input parameter to resend a DAN. When the operator selects the OK button on the form, the form invokes ldt_SendDAN.

- **Manage Files...** activates an Oracle form requesting retention or deletion of files for a specific contact sequence identifier. When the operator selects the OK button on the form, the form updates the Marked_For_Retention field in the LDT_Output_File_State_Info table.

13.3.1.7 REPORTS Menu Item

Three submenus are available under REPORTS:

- **Data Receive Summary...** activates an Oracle form requesting a contact sequence identifier. When the operator selects the OK button on the form, the form invokes rdc_GenDataRcvSumReport report file to generate the report.
- **LPS QA...** activates an Oracle form requesting a contact sequence identifier. When the operator selects the OK button on the form, the form invokes mac_GenQAReport report file to generate the report.
- **File Transfer Summary...** activates an Oracle form requesting contact sequence identifiers. When the operator selects the OK button on the form, the form invokes ldt_GenFTS report file to generate the reports.

13.3.2 Database Forms and Reports

Database forms and reports will be used by the operational users to enter input parameters to activate LPS tasks and view and modify data in the database. After a user selects a menu item from the main menu, a form is displayed. A user can select to have the report displayed on the screen or output to a file.

The following form descriptions contain the form’s name, its purpose, and its associated database table names:

- Form name: LPS String Configuration
  - Purpose: Provide operational personnel with the capability to view, insert, update, and delete the LPS string configuration.
  - Database table(s): LPS_Configuration
- Form name: Data Processing Thresholds
  - Purpose: Provide operational personnel with the capability to view, insert, update, and delete the data processing thresholds.
  - Database table(s): Valid_RDP_Thres, Valid_MFP_Thres, and Valid_PCD_Thres
- Form name: Data Processing Parameters
  - Purpose: Provide operational personnel with the capability to view, insert, update, and delete the data processing parameters
  - Database table(s): Valid_PCD_Parms, Valid_MFP_Parms, Valid_CCSDS_Parms, Valid_Scene_Parms, and Valid_Band_Parms

- Form name: Contact Schedules
  - Purpose: Provide operational personnel with the capability to view, insert, update, and delete contact schedule.
  - Database table(s): Contact_Schedules and LPS_Configuration
- Form name: Ingest IAS Parameters
  - Purpose: Provide operational personnel with the capability to view and update IAS parameters.
  - Database table(s): Valid_MFP_Parms and Valid_Scene_Parms
- Form name: Start Data Capture
  - Purpose: Provide operational personnel with the capability to start raw data capture.
  - Database table(s): Contact_Schedules
- Form name: Stop Data Capture
  - Purpose: Provide operational personnel with the capability to stop raw data capture.
  - Database table(s): Contact_Schedules
- Form name: Start Copy Data to Tape
  - Purpose: Provide operational personnel with the capability to copy data to tape.
  - Database table(s): RDC_Acct
- Form name: Stop Copy Data to Tape
  - Purpose: Provide operational personnel with the capability to stop copying data to tape.
  - Database table(s): RDC_Acct
- Form name: Start Copy Data from Tape
  - Purpose: Provide operational personnel with the capability to copy data from tape to the restage disk.
  - Database table(s): RDC_Acct
- Form name: Stop Copy Data to Tape
  - Purpose: Provide operational personnel with the capability to stop copying data from tape to the restage disk.
  - Database table(s): RDC_Acct

- Form name: Start Data Processing
  - Purpose: Provide operational personnel with the capability to start data processing tasks (raw data processing, major frame processing, PCD processing, and image data processing).
  - Database table(s): RDC_Acct
- Form name: Stop Data Processing
  - Purpose: Provide operational personnel with the capability to stop data processing tasks (raw data processing, major frame processing, PCD processing, and image data processing).
  - Database table(s): RDC_Acct
- Form name: Display LPS Journal File
  - Purpose: Provide operational personnel with the capability to display the entire or a specified range of the LPS Journal file.
  - Database table(s): None
- Form name: Display Operation Messages
  - Purpose: Provide operational personnel with the capability to display latest operation messages with the specified severity level.
  - Database table(s): None
- Form name: Enable/Disable DAN Transfer
  - Purpose: Provide operational personnel with the capability to enable or disable DAN transfer.
  - Database table(s): Process_Id
- Form name: Resend DAN
  - Purpose: Provide operational personnel with the capability to resend DANs to EDC DAAC for a specific contact period.
  - Database table(s): LDT_File_Set_Info and LDT_DAN_Transfer_State
- Form name: Manage Files
  - Purpose: Provide operational personnel with the capability to retain or delete files for a specific contact sequence identifier.
  - Database table(s): LDT_File_Set_Info and Sub_Intv

The following report descriptions contain the report’s name, its purpose, and its associated database table names:

- Report name: Data Receive Summary
  - Purpose: Display data receive accounting information, including contact sequence identifier, LPS hardware string identifier, LGS channel identifier, contact start time, contact stop time, and received data volume.
  - Database table(s): RDC_Acct
- Report name: LPS QA
  - Purpose: Display LPS Q&A data information, including subinterval information and CADU information.
  - Database table(s): Sub_Intv, RDP_Acct, RDC_Acct, MFP_Acct, IDP_Acct, LPS_Configuration, and Valid_CCSDS_Parms
- Report name: File Transfer Summary
  - Purpose: Display daily file transfer information, including data types, number of files available for transfer, and number of files that have been transmitted.
  - Database table(s): LDT_File_Set_Info, LDT_File_Group_Info, LDT_File_Info, and Sub_Intv

13.4 Operational Support

The user interface provides menu options that allow operations personnel to perform LPS operational activities. The following steps support normal operations:

1. **Starting the user interface** – Before starting the user interface, the operator must set up the operating environment. The operating environment includes a user account in the database and a GUI environment for Oracle CDE tools.

   To start the user interface after the environment is configured, invoke the alias

   % start_lps_ui string**n** (**n** = 1, 2, 3 or 4)

   from the command line prompt. The main menu is displayed.

2. **Set Up LPS String Configuration** – Selecting “LPS String Configuration...” from the SETUP menu displays the LPS String Configuration form. This form allows operator to update the LPS string configuration in the database.

3. **Set Up Data Processing Thresholds** – Selecting “Data Processing Threshold...” from the SETUP menu displays the Data Processing Thresholds form. This form allows operator to update data processing thresholds in the database.

4. **Set Up Data Processing Parameters** – Selecting “Data Processing Parameters...” from the SETUP menu displays the Data Processing Parameters form. This form allows operator to update data processing parameters in the database.

5. **Set Up Contact Schedules –** Selecting "Contact Schedules..." from the SETUP menu displays the Contact Schedules form. This form allows operator to insert and update contact schedules in the database.

6. **Starting Operation Messages Monitor Window –** Selecting "Display Operation Messages..." from the MONITOR menu displays the Display Operation Messages form. This form allows operator to select the severity level of messages to be displayed on the operation messages window. After the operator selects the OK button on the form, an X Window is displayed on the screen. All of the latest operation messages with the specified severity level will be displayed on the window.

7. **Starting Raw Data Capture –** Selecting "Start Data Capture..." from the CONTROL menu displays the Start Data Capture form. The earliest contact schedule is automatically displayed on the form. The raw data capture task is started after the operator verifies the schedule time and presses the OK button.

8. **Starting Copy Data to Tape –** Selecting "Start Copy Data to Tape..." from the CONTROL menu displays the Start Copy Data to Tape form to prompt for the contact sequence identifier. After the operator enters the contact sequence identifier and selects the OK button on the form, the form invokes rdc_Save.

9. **Stopping Copy Data to Tape –** Selecting "Stop Copy Data to Tape..." from the CONTROL menu displays the Stop Copy Data to Tape form to prompt for the contact sequence identifier. After the operator enters the contact sequence identifier and selects the OK button on the form, the form invokes rdc_StopSaveRestage.

10. **Starting Copy Data from Tape –** Selecting "Start Copy Data from Tape..." from the CONTROL menu displays the Start Copy Data from Tape form to prompt for the contact sequence identifier. After the operator enters the contact sequence identifier and selects the OK button on the form, the form invokes rdc_RestageCptr.

11. **Stopping Copy Data from Tape –** Selecting "Stop Copy Data from Tape..." from the CONTROL menu displays the Stop Copy Data from Tape form to prompt for the contact sequence identifier. After the operator enters the contact sequence identifier and selects the OK button on the form, the form invokes rdc_StopSaveRestage.

12. **Starting Data Processing –** Selecting "Start Data Processing..." from the CONTROL menu displays the Start Data Processing form to prompt for the contact sequence identifier. The data processing task is invoked after the operator enters the contact sequence identifier and selects the OK button on the form. The status of data processing is displayed on the operation messages window.

13. **Generating Data Receive Summary Report –** After raw data capture is completed, selecting "Data Receive Summary..." from the REPORTS menu displays the Data Receive Summary form to prompt for the contact sequence identifier. After the operator enters the contact sequence identifier and selects the OK button on the form, the form invokes a report file to generate the report.

14. **Generating LPS Q&A Report –** Selecting "LPS QA..." from the REPORTS menu displays the LPS QA form to prompt for the contact sequence identifier. After the operator

enters the contact sequence identifier and selects the OK button on the form, the form invokes a report file to generate the report.

15. **Generating File Transfer Summary –** Selecting “File Transfer Summary...” from the REPORTS menu displays the File Transfer Summary form to prompt for the contact sequence identifier. After the operator enters the contact sequence identifier and selects the OK button on the form, the form invokes a report file to generate the report.

16. **Reprocessing LPS Data –** If reprocessing data is available in 30-day storage, selecting “Start Copy Data from Tape...” from the CONTROL menu displays the Start Copy Data from Tape form to prompt for the contact information. After the operator enters the contact information and selects the OK button, the data on the tape is copied from the tape to the restage disk. Selecting “Start Data Processing...” from the CONTROL menu starts data processing.

17. **Shut Down User Interface –** Selecting SHUTDOWN from the main menu exits the user interface. All user interface windows are deleted from the display after all the processes are completed and terminated.

Appendix A. Requirements Traceability

This appendix presents the LPS requirements traceability. Table A–1 shows the mapping between the lowest level processes in the DFDs and the LPS software modules.

Table A–1. Software Requirements to Software Module Traceability

F&PS Requirement	Software Require-ment No.	Software Requirement Name	LPS Software Module
3.1.4, 3.1.6, 3.1.11, 3.2.1, 3.3.1.1, 3.3.1.2, 3.3.1.3, 3.3.1.4, 3.3.1.12, 3.3.2.13, 3.3.4.12	1.1	Receive Raw Wideband Data	rdc_Capture rdc_UpdRDCAcct
3.1.4, 3.3.1.6, 3.3.1.7	1.2	Save Raw Wideband Data	rdc_Save
3.3.1.9	1.3	Restage Raw Wideband Data	rdc_RestageCptr
3.3.1.1, 3.3.1.2, 3.3.1.3, 3.3.1.4	1.4	Delete Raw Wideband Data	rdc_DeleteRDCFiles
3.3.1.5, 3.3.1.6	1.5	Replay Raw Wideband Data	rdc_Transmit
3.3.1.10	1.6	Generate Media Label	rdc_GenLabel
	2.1	Synchronize CCSDS Frame	
3.1.4, 3.1.6, 3.1.7, 3.3.2.2, 3.3.2.3, 3.3.2.4, 3.3.2.13, 3.3.2.26, 3.3.2.27, 3.3.4.13, 3.3.6.7	2.1.1	Perform SCLF Synchronization	rdp_MainExtractCADU rdp_MainFSync rdp_MainTranToShared
3.1.4, 3.3.2.6	2.1.2	Align Bytes	rdp_MainExtractCADU
3.1.4, 3.1.6, 3.3.2.3, 3.3.2.5, 3.3.2.13	2.1.3	Deinvert Data	rdp_MainExtractCADU
3.1.4, 3.3.2.7	2.1.4	Perform PRN Decode	rdp_MainExtractCADU
	2.2	Process CCSDS Grade 3	
3.1.4, 3.1.6, 3.1.7, 3.3.2.1, 3.3.2.8, 3.3.2.13, 3.3.2.26, 3.3.2.27, 3.3.4.13, 3.3.6.7	2.2.1	Perform CRC Check	rdp_MainValidateCADU rdp_MainGenerateOutput
3.1.4, 3.1.6, 3.1.7, 3.3.2.1, 3.3.2.8, 3.3.2.13, 3.3.2.26, 3.3.2.27, 3.3.4.13, 3.3.6.7	2.2.2	Perform RS_EDAC Check	rdp_MainValidateCADU rdp_MainGenerateOutput
3.1.4, 3.1.6, 3.1.7, 3.3.2.9, 3.3.2.9.1, 3.3.2.10, 3.3.2.13, 3.3.2.26, 3.3.2.27, 3.3.4.13, 3.3.6.7	2.3	Decode BCH	rdp_BCHDecode rdp_MainGenerateOutput
3.1.4, 3.3.2.11	2.4	Annotate VCID Change	rdp_MainValidateCADU

F&PS Requirement	Software Requirement No.	Software Requirement Name	LPS Software Module
3.3.2.11, 3.3.2.12, 3.3.2.19	3.1	Identify VCDUs	mfp_MainIdentifyMjfSet
3.3.2.28, 3.3.4.16	3.2	Extract PCD	mfp_MainPcdStatusProc
	3.3	Parse Major Frame	
3.3.2.14, 3.3.2.15	3.3.1	Identify Major Frames	mfp_MainValidateMjf
3.1.7, 3.3.2.26	3.3.2	Extract Major Frame Time	mfp_MainValidateMjf
3.1.7, 3.3.2.26, 3.3.2.27, 3.3.6.7	3.3.3	Collect VCDU Quality and Accounting	mfp_MainValidateMjf
3.3.2.11, 3.3.2.23	3.3.4	Determine Subintervals	mfp_MainDetermineSub
	3.4	Generate Band Data	mfp_MainBandGen
3.3.2.14, 3.3.2.16, 3.3.2.17, 3.3.2.18	3.4.1	Deinterleave and Reverse Bands	mfp_MainBandGen
3.3.2.19, 3.3.2.22	3.4.2	Align Bands	mfp_MainBandGen
	3.5	Extract Calibration and MSCD	mfp_L0RFilesGen
3.1.5, 3.3.2.24, 3.3.2.25, 3.3.2.28, 3.3.4.12	3.5.1	Create MSCD File	mfp_L0RFilesGen
3.1.5, 3.3.2.24, 3.3.2.25, 3.3.4.12	3.5.2	Create Calibration File	mfp_L0RFilesGen
3.3.2.14, 3.3.2.20	3.5.3	Extract MSCD Data	mfp_L0RFilesGen
3.3.2.14, 3.3.2.21	3.5.4	Extract Calibration Data	mfp_L0RFilesGen
3.1.6, 3.1.7, 3.3.2.13, 3.3.2.26, 3.3.4.13	3.6	Collect Quality and Accounting	mfp_MainQASubGen
	4.1	Assemble PCD Frames	
3.3.4.1	4.1.1	Locate PCD Sequences	pcd_MainDeterminePcdWord
3.3.4.1	4.1.2	Sync on Unpacked PCD	pcd_MainDeterminePcdWord
3.3.4.1, 3.3.4.15	4.1.3	Pack PCD	pcd_MainDeterminePcdWord
3.3.4.1, 3.3.4.14	4.1.4	Assemble PCD Minor Frames	pcd_MainBuildCycle
3.3.4.3, 3.3.4.14	4.1.5	Assemble PCD Major Frames	pcd_MainBuildCycle
3.3.4.3, 3.3.4.14	4.1.6	Assign PCD Major Frame Times	pcd_MainBuildCycle

F&PS Requirement	Software Requirement No.	Software Requirement Name	LPS Software Module
3.3.6.7	4.1.7	Issue Missing Words Alarm	pcd_MainBuildCycle
3.3.6.7	4.1.8	Issue Majority Vote Failure Alarm	pcd_MainBuildCycle
3.3.4.5, 3.3.4.12, 3.3.4.14, 3.3.4.15	4.2	Summarize PCD Quality	pcd_MainCreatePcdFile
3.3.4.12	4.3	Extract Bands Present	pdc_MainExtractCycleInfo
3.3.4.7, 3.3.4.15, 3.3.4.16	4.4	Extract Spacecraft Position	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.15, 3.3.4.16	4.5	Validate Spacecraft Position	pcd_MainDetermineScenes
	4.6	Identify WRS Scenes	
3.3.4.7, 3.3.4.16	4.6.1	Compute Look Points	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16, 4.3.5	4.6.2	Locate Scene Centers	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16, 4.3.5	4.6.3	Compute Corner Positions	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16, 4.3.5	4.6.4	Compute Horizontal Display Shift	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16	4.6.5	Calculate Sun Position	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16	4.6.6	Assemble Scene Messages	pcd_MainDetermineScenes
3.3.2.29, 3.3.4.12, 3.3.4.16	4.6.7	Assemble Scene Metadata	pcd_MainDetermineScenes
3.3.2.29, 3.3.4.16	4.6.8	Report CAL Door Activity	pcd_MainDetermineScenes
3.3.4.4	4.7	Create PCD File	
3.3.4.4	4.7.1	Convert ADS to EU	pcd_MainCreatePcdFile
3.3.4.4	4.7.2	Convert ADS Temperature to EU	pcd_MainCreatePcdFile
3.3.4.4	4.7.3	Convert Gyro to EU	pcd_MainCreatePcdFile
3.3.4.4	4.7.4	Convert Gyro Drift to EU	pcd_MainCreatePcdFile
3.1.5, 3.3.2.24, 3.3.2.25, 3.3.4.2, 3.3.4.4	4.7.5	Create PCD Output Files	pcd_MainCreatePcdFile

F&PS Requirement	Software Require- ment No.	Software Requirement Name	LPS Software Module
3.3.4.4, .3.4.12	4.7.6	Create PCD File Names	pcd_MainCreatePcdFile
	5.2	Generate Browse File	
3.3.3.1, 3.3.3.3, 3.3.3.4	5.2.1	Correct and Contrast Stretch Image	idp_Browse
3.1.5, 3.3.2.24, 3.3.3.1, 3.3.3.3, 3.3.3.4, 3.3.3.5, 3.3.4.12, 4.3.4	5.2.2	Reduce Image by Wavelets	idp_Browse
3.1.5, 3.3.2.24, 3.3.2.25, 3.3.4.12, 3.3.4.16	5.3	Generate Band File	idp_Band
	5.4	Perform ACCA	
3.3.4.8, 3.3.4.9	5.4.1	Collect Scene Data	idp_ACCA
3.3.4.8, 3.3.4.9, 3.3.4.10, 3.3.4.16	5.4.2	Generate Cloud Cover Assessment	idp_ACCA
3.3.6.10	5.5	Generate Moving Window Display	idp_mwd
3.1.11, 3.3.5.5, 3.3.5.6, 3.3.6.1, 3.3.6.4, 3.3.6.5	6.1	Process LPS Directive	mac_ui_MainMenu
3.1.5, 3.3.2.24, 3.3.4.11, 3.3.4.12, 3.3.4.13, 3.3.4.14, 3.3.4.16	6.2	Generate Metadata	mac_MetaDataGen
3.1.11, 3.3.6.7	6.3	Report LPS Status	mac_ui_MainMonitor
3.2.4, 3.3.4.12, 3.3.6.1	6.4	Ingest CAL File	mac_ui_MainControl
3.1.14, 3.3.6.1, 3.3.6.6, 3.3.6.9	6.5	Modify L0R Parameters and Thresholds	mac_ui_MainSetup
3.2.3, 3.3.6.1	6.6	Ingest Contact Schedule	mac_ui_MainSetup
3.1.10.1, 3.1.10.2, 3.3.6.8	6.7	LPS System Control	mac_ui_MainFileMgt mac_yu_MainControl
3.1.10.4, 3.1.10.5	6.8	Monitor System Faults	mac_ui_MainMonitor
	6.9	Validate Parameters	
3.3.6.1	6.9.1	Validate RDP Parameters	mac_ui_MainSetup
3.3.6.1	6.9.2	Validate MFP Parameters	mac_ui_MainSetup
3.3.6.1	6.9.3	Validate IDP Parameters	mac_ui_MainSetup
3.3.6.1	6.9.4	Validate PCD Parameters	mac_ui_MainSetup

F&PS Requirement	Software Require-ment No.	Software Requirement Name	LPS Software Module
3.3.6.1	6.9.5	Validate RDC LDT Parameters	mac_ui_MainSetup
	6.10	Display or Print LPS Report	
3.3.1.10	6.10.1	Generate Data Receive Summary Report	mac_ui_MainReports
3.1.6, 3.1.7, 3.3.2.13, 3.3.2.26, 3.3.6.2, 3.3.6.3, 3.3.6.4, 3.3.6.4.1	6.10.2	Generate Level 0R QA Report	mac_ui_MainReports
3.3.5.7, 3.3.6.5, 3.3.6.5.1	6.10.3	Generate File Transfer Summary Report	mac_ui_MainReports
3.2.2, 3.3.5.1	7.1	Generate DAN	ldt_CreateDAN
3.2.2, 3.3.5.1	7.2	Send DAN	ldt_SendDAN
3.3.5.3	7.3	Receive DDN	ldt_RcvDDN
3.2.2, 3.3.5.1	7.4	Resend DANs	ldt_RsndSuspDANs
3.3.5.4, 3.3.5.5	7.5	Delete LPS Files	ldt_DeleteFiles
3.3.5.4, 3.3.5.6	7.6	Retain LPS Files	ldt_RetainFiles
3.2.2, 3.3.5.1	7.7	Control Send DAN	ldt_RsndSuspDANs

Appendix B. LPS Software Size Estimates

Release 2.0 counts are as follows:

Subsystem	No. of Units	DSI	Comment Lines	Blank Lines	Total Lines
COTS	10	812	1046	125	1983
DB	11	471	990	121	1582
Global	72	3807	6906	1194	11907
IDPS	147	9426	14848	3737	28001
LDTS	87	6077	10481	1651	18209
MACS	63	5531	6929	1389	13849
MFPS	79	9236	11931	2641	23808
PCDS	86	6972	9889	1911	18772
RDCS	72	4333	8357	1224	13914
RDPS	45	5069	5077	762	10908
UI	37	14866	146	9152	24164
Total	709	66600	76600	23907	167097

Appendix C. LPS RMA Analysis

C.1 Objective

The objective of this analysis is to ensure that the LPS hardware configuration, consisting of five independent strings, adequately meets and/or exceeds the LPS operational availability requirements of 0.96 as defined and specified in the LPS F&PS.

C.2 Assumptions

The LPS reliability, maintainability, availability (RMA) analysis is based on the following assumptions:

- The LPS consists of five logically independent and identical strings.
- A maximum of four strings are required at all times to support LPS operations.
- The hardware configuration for each LPS processing string is as shown in Figure 2–6 of the SDS.
- The following effort (maximum times) is required to switch out the malfunctioning string and switch in the fifth string to restore LPS operations:
  - Time to detect LPS processing string problem: 5 minutes
    a. Time to repossess fifth string from LPS development/test: 10 minutes (It may take longer than 10 minutes if the fifth string requires reconfiguration to support LPS operations.)
    b. Time to repair failed string if spares available at site: 120 minutes
    c. Time to repair failed string if spares not available at site: 8 minutes (includes expected response time for vendor maintenance)
  - Time to bring up LPS programs on fifth string: 15 minutes:
  - Time to test LPS operations of fifth string: 10 minutes
  - Time to test the fifth string with LGS and EDC DAAC: 15 minutes

C.3 LPS RMA Analysis

LPS RMA analysis were performed in two steps. single string RMA calculations and all five strings' (LPS) RMA calculations. These calculations are shown in Tables C–1 and C–2, respectively.

The first step is used to calculate the mean time between failure (MTBF) and availability of a single LPS string. The first step calculations are used to calculate the overall availability of a single string using the MTBF data available on all the hardware in a string that must be always be operational to make the whole string operational. Redundant hardware items, if any, are also considered in this analysis.

The second step is performed for two configurations of the LPS, a four-string LPS and a five-string LPS. Both configurations of the LPS use four strings for LPS operations. The fifth string, if any, is used as a backup to the four operational strings.

Detailed information on performing RMA analysis, definitions, and calculations are provided in *SSDM Guidelines*.

C.4 Results of RMA Analysis

The following results are derived from RMA analysis of the LPS hardware and architecture (string configurations):

- A four-string LPS can meet the LPS RMA requirements [operational availability of 0.96 and mean time to restore (MTTRes) of 4 hours] provided that the EDC has adequate spares and maintenance personnel available onsite to repair the failed string within 3 hours. This would allow EDC maintenance personnel sufficient time to complete all system checkout activities and restore LPS operations during the fourth hour.

- A four-string LPS will not meet the MTTRes requirement of 4 hours if no spares and/or maintenance personnel are available at the EDC to repair the failed string. Any vendor maintenance arranged must respond and repair the failed string within 3 hours (which is highly unlikely) to restore LPS operations.

- A five-string LPS can meet the operational availability requirement of 0.96 and the MTTRes requirement of 4 hours, provided that the fifth string is returned to LPS operations with minutes of issuing a request to the LPS test and development groups. This requires that the LPS fifth string, when returned, should be properly configured to replace the failed string and to support LPS operations. If a major reconfiguration of the fifth string is required, this may jeopardize restoring LPS operations within the required 4 hours of MTTRes time.

Table C–1. Results of LPS RMA Analysis
(Four of Five Strings to LPS Operations) (1 of 2)

Operational availability = 0.9979058						
MTTRes = 55.00 minutes						
RMA Modeling Calculations						
Downtime measurement period = 10,000 hours						
or total time						
Item Name	**Item MTBF**	**Items Required for Operations**	**Items Available to Operations**	**Downing Events Over 10,000 Hours**	**Switchover Time or Item MTTR (min)**	**Average Downtime Over 10,000 Hours (min)**
LPS string	1751	4	5	22.85	55.00	1256.54
Totals				22.85		1256.54
RMA Calculation Assumptions						
Four of five strings required for operation; restore LPS by switchover to the fifth string.						
Orbit period: 90 minutes						
Contact period: 14 minutes						
Available time for restore: 76 minutes						
Time to detect string problem: 5 minutes						
Time to obtain string: 10 minutes (from test/development)						
Time to bring up LPS: 15 minutes						
Time to test LPS: 10 minutes						
Time to test with LGS and EDC DAAC: 15 minutes						
Switchover/restore time: 55 minutes						
Time to wait for Landsat 7 pass: 21 (extra minutes available)						

Table C–1. Results of LPS RMA Analysis
(Four of Five Strings to LPS Operations) (2 of 2)

<table>
<tr><td colspan="7">Operational availability = 0.9937173</td></tr>
<tr><td colspan="7">MTTRes = 165.00 minutes</td></tr>
<tr><th colspan="7">RMA Modeling Calculations</th></tr>
<tr><td colspan="7">Downtime measurement period = 10,000 hours</td></tr>
<tr><td colspan="7">or total time</td></tr>
<tr><th>Item Name</th><th>Item MTBF</th><th>Items Required for Operations</th><th>Items Available to Operations</th><th>Downing Events Over 10,000 Hours</th><th>Switchover Time or Item MTTR (min)</th><th>Average Downtime Over 10,000 Hours (min)</th></tr>
<tr><td>LPS string</td><td>1751</td><td>4</td><td>4</td><td>22.85</td><td>165.00</td><td>3769.61</td></tr>
<tr><td>Totals</td><td></td><td></td><td></td><td>22.85</td><td></td><td>3769.61</td></tr>
<tr><th colspan="7">RMA Calculation Assumptions</th></tr>
<tr><td colspan="7"><b>All four strings to operation; LPS restored by repairing failed string.</b></td></tr>
<tr><td colspan="7">Orbit period: 90 minutes</td></tr>
<tr><td colspan="7">Contact period: 14 minutes</td></tr>
<tr><td colspan="7">Available time for restore: 76 minutes</td></tr>
<tr><td colspan="7">Time to detect string problem: 5 minutes</td></tr>
<tr><td colspan="7">Time to repair failed string: 120 minutes (spares available onsite)</td></tr>
<tr><td colspan="7">Time to bring up LPS: 15 minutes</td></tr>
<tr><td colspan="7">Time to test LPS: 10 minutes</td></tr>
<tr><td colspan="7">Time to test with LGS and EDC DAAC: 15 minutes</td></tr>
<tr><td colspan="7">Switchover/restore time: 165 minutes</td></tr>
<tr><td colspan="7">Time to wait for Landsat 7 pass: -89 (no time to capture next pass; two passes lost)</td></tr>
</table>

Table C–2. LPS Single-String Configuration – RMA Calculations for Series Components (1 of 2)

	(CM)			**(1/SK)**
Configuration/ Component Name	**Component MTBF (CM)**	**K = (1/CM)**	**Sum of all Ks (SK)**	**Configuration MTBF**
LPS string			0.00057115	1751
Challenge L power board	310000	3.22581E-06		
CPU board	35000	2.85714E-05		
Memory board	35000	2.85714E-05		
16-MB SIMM 1	2400000	4.16667E-07		
16-MB SIMM 2	2400000	4.16667E-07		
16-MB SIMM 3	2400000	4.16667E-07		
16-MB SIMM 4	2400000	4.16667E-07		
16-MB SIMM 5	2400000	4.16667E-07		
16-MB SIMM 6	2400000	4.16667E-07		
16-MB SIMM 7	2400000	4.16667E-07		
16-MB SIMM 8	2400000	4.16667E-07		
POWERchannel I/O board	35000	2.85714E-05		
S HIO module	35000	2.85714E-05		
FDDI board (VME)	35000	2.85714E-05		
Serial-to-parallel board	35000	2.85714E-05		
DSP board	35000	2.85714E-05		
2-GB disk	500000	0.000002		
26B DAT	40000	0.000025		
CD-ROM	250000	0.000004		
8-GB RAID disk 1	500000	0.000002		
8-GB RAID disk 2	500000	0.000002		
8-GB RAID disk 3	500000	0.000002		
8-GB RAID disk 4	500000	0.000002		
8-GB RAID disk 5	500000	0.000002		
RAID controller	40000	0.000025		
RAID SCSI module	35000	2.85714E-05		
RAID power assembly	310000	3.22581E-06		
16-GB RAID disk 1	500000	0.000002		
16-GB RAID disk 2	500000	0.000002		
16-GB RAID disk 3	500000	0.000002		
16-GB RAID disk 4	500000	0.000002		
16-GB RAID disk 5	500000	0.000002		
RAID controller	40000	0.000025		
RAID SCSI module	35000	2.85714E-05		

Table C–2. LPS Single-String Configuration – RMA Calculations for Series Components (2 of 2)

	(CM)			(1/SK)
Configuration/ Component Name	**Component MTBF (CM)**	**K = (1/CM)**	**Sum of all Ks (SK)**	**Configuration MTBF**
RAID power assembly	310000	3.22581E-06		
7.5-Mbps recorder	10000	0.0001		
LPS console/workstation	10000	0.0001		

Appendix D. Performance Analysis

Initial performance analysis currently shows that LPS performs at approximately 5 megabits per second for Format 2 data. The performance specification is 7.5 megabits per second. ACCA processing further delays Format 1 processing. Analysis is underway for improving the performance.

Appendix E. LPS Algorithms

This appendix presents the algorithms that have been implemented for the LPS:

- Clarification of WRS scene corners
- LPS WRS horizontal display shift algorithm
- Sun azimuth and elevation algorithms
- WRS scene center and corner algorithms

E.1 Clarification of WRS Scene Corners

E.1.1 Description of Problem

In August 1996, Ms. Minnie Wong reported a problem with the results of the WRS Scene Determination algorithm. Some of the WRS corners produced by Mr. Rich McIntosh's program were found to be upside-down, i.e., the leading edge latitude was less than the trailing edge latitude. This problem was referred to Mr. McIntosh, who corrected the problem by fixing a flag that did not correctly switch at the spacecraft ascending and descending transitions. The WRS algorithm now correctly reports the trailing and leading corners of scenes for ascending and descending orbits. Additionally, there was some ambiguity regarding the mapping of leading and trailing corners, produced by the WRS algorithm, to those reported in the LPS metadata (upper and lower corners). It was also not clear whether this mapping, if any was required, remained the same or changed for the trailing and leading corners of ascending and descending scenes. Additional clarification on scene corners were required to clarify their reporting in the LPS metadata file.

E.1.2 Definitions

To clarify the reporting of scene corners in the LPS metadata, the following definitions are provided in consultations with Mr. McIntosh and Mr. Rich Irish:

- Trailing Edge – Edge of a scene behind the spacecraft while it is ascending or descending. This edge is determined by the first scan in a scene.
- Leading Edge – Edge of a scene in front (ahead) of the spacecraft while it is ascending or descending. This edge is determined by the last scan in a scene.
- Left Corner – Corner to the left of the spacecraft, either ascending or descending.
- Right Corner – Corner to the right of the spacecraft, either ascending or descending.

Figure E–1 illustrates the context of leading and trailing edges and left and right corners for ascending and descending spacecraft.

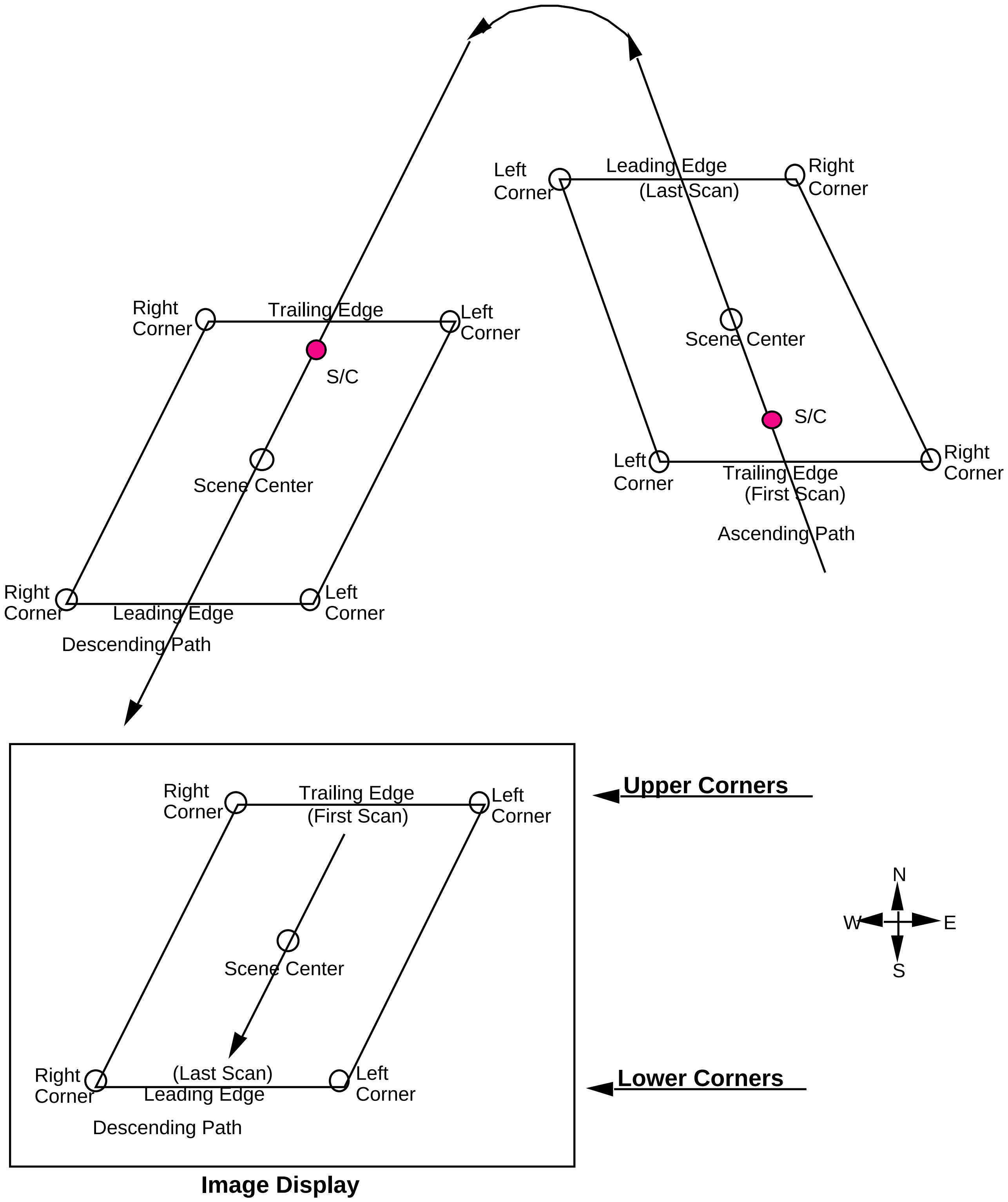


Figure E–1. WRS Scene Corners Context

E.1.3 Observations

The following observations are made from Figure E–1 and using the definitions in Section E.1.2:

- Scene corners to spacecraft context is fixed regardless of whether the spacecraft is ascending or descending. Left corners are always to the left of the spacecraft, and right corners are to the right of the spacecraft.

- When acquired scan data is displayed in the order of its arrival, the trailing edge (first image observation and captured scan) is always displayed on top (upper edge of scene). This is true for both ascending and descending scene (scans). Similarly, the leading edge (last image observation and captured scan) is always displayed on bottom (lower edge of scene).

- Trailing (edge) corners, which are obtained from WRS scene calculations, determine upper corners, while leading (edge) corners determine the lower corners of a scene (image display) reported in the LPS metadata.

- WRS algorithm determined left corners (from the left of spacecraft) are displayed on the right side of the image display and the image observer. Similarly, right corners (from the right of spacecraft) are displayed on the left side of the image display and the image observer. The terms “left” and “right” mean the same with respect to the spacecraft. They are reversed when used in context of the image display and/or the image observer.

- For all corner contexts fixed to the spacecraft, a simple mapping of trailing and leading edge corners, determined by the WRS algorithm, to upper and lower corners, reported in LPS metadata, is required. This mapping is fixed for both the ascending and descending paths and is as follows:

  - Trailing (edge) left corner = upper left corner of scene

  - Trailing (edge) right corner = upper right corner of scene

  - Leading (edge) left corner = lower left corner of scene

  - Leading (edge) right corner = lower right corner of scene

E.1.4 Conclusion

Other than providing a simple mapping between trailing and leading corners to upper and lower corners, respectively, there is no other impact on the LPS PCDS software design.

E.2 LPS WRS Horizontal Display Shift Algorithm

E.2.1 Problem Statement

Given the locations of two nominal scene centers and the actual scene center, find the horizontal display shift (HDS) in kilometers on the geoid. The HDS is defined as the perpendicular distance of the actual scene center from the nominal scene center track.

E.2.2 Solution

The input required for this function are the latitudes and longitudes of the actual scene center and two nominal scene centers and the spacecraft velocity vector at the time of the actual scene center. Before finding the HDS, it is necessary to find the perpendicular angular distance on the unit sphere from the actual scene center to the nominal track. This is given by

$$d = \cos^{-1}[\vec{N} \;\; \hat{R}] - \frac{\pi}{2}$$

Where:

$\vec{R} = (\alpha_a, \delta_a)$ is the actual scene center

$\hat{C}_1 = (\alpha_1, \delta_1)$ is the nominal scene center nearest the actual center ($\text{Path}_1, \text{Row}_1$)

$\hat{C}_2 = (\alpha_2, \delta_2)$ is the previous nominal scene center ($\text{Path}_1, \text{Row}_1 - 1$)

$\vec{N} = (\alpha_{\text{normal}}, \delta_{\text{normal}})$ is the normal vector formed from $\hat{C}_1$ and $\hat{C}_2$

(α, δ) is longitude and geocentric latitude

The vectors to each of the three scene centers are formed from the longitude and latitude as follows (using the appropriate subscripts for each):

$$\begin{bmatrix} \cos(\delta)\cos(\alpha) \\ \cos(\delta)\sin(\alpha) \\ \sin(\delta) \end{bmatrix}$$

The unit normal is obtained from

$$\hat{N} = \frac{\hat{C}_2 \times \hat{C}_1}{\left\| \hat{C}_2 \times \hat{C}_1 \right\|}$$

Note that d can be either positive or negative. The sign of d is a function of the relative east/west position of the actual scene center to that of the nominal track. This relationship can be expressed as a Boolean function, East_True, using the sign of the z component of velocity, V_z, and the sign of d.

$$\text{East\_True} = (1 + \text{sign}(V_z) \;\; \text{sign}(d))/\,2$$

East_True = 1 means that the actual scene center is east of the nominal track; East_True = 0 implies that it is west of the nominal track.

Using the radius of the Earth at the latitude of the actual scene center, the value of the HDS in kilometers on the Earth’s surface can be calculated:

$$r(\delta_a) = \frac{a\ \ b}{\sqrt{a^2 \sin^2(\delta_a) + b^2 \cos^2(\delta_a)}}$$

$$HDS = r(\delta_a)\ \ d$$

Where:

a is the equatorial radius of the Earth

b is the polar radius of the Earth

E.3 Sun Azimuth and Elevation Algorithms

This section describes and analyzes a Sun azimuth and elevation computation algorithm intended for implementation in the LPS. The section provides the following information:

- Summary of algorithm’s input data
- Description of algorithm
- Analysis of algorithm’s computational complexity
- Estimate of DSIs required to implement the algorithm
- Source listings for FORTRAN subroutines that compute the geocentric inertial (GCI) Sun vector and Greenwich hour angle (GHA).

E.3.1 Input Data and Their Sources

The Sun azimuth and elevation computation algorithm requires the input listed.

- Latitude (*lat*) and longitude (*lon*) of the WRS scene center.
- Spacecraft time (*t*) at which the WRS scene center data was obtained.

All values are computed during WRS scene identification.

E.3.2 Algorithm Description

The problem to be addressed is as follows: Given the latitude, longitude, and spacecraft time of the WRS scene center, compute the Sun azimuth (AZ) and elevation (EL) at the ground point. The algorithm is as follows:

1. Compute the GHA (right ascension of Greenwich) at time *t* using subroutine JGHAX (attached).
2. Compute the GCI Sun vector ($\vec{R}_{Sun}$) at time *t* using subroutine SOL (attached).
3. Define an Earth-fixed coordinate system centered at the *lat* and *lon* of interest with coordinate axes pointing north (N), east (E), and local vertical (V):

$$\widehat{V} = \begin{vmatrix} \cos(lon)\cos(lat) \\ \sin(lon)\cos(lat) \\ \sin(lat) \end{vmatrix}$$

$$\widehat{E} = \left| \begin{matrix} -\sin(lon) \\ \cos(lon) \\ 0 \end{matrix} \right. .$$

$$\widehat{N} = \begin{vmatrix} -\cos(lon)\sin(lat) \\ -\sin(lon)\sin(lat) \\ \cos(lat) \end{vmatrix}$$

4. Compute the transformation matrix from Earth-fixed to GCI:

$$[G] = \begin{vmatrix} \cos(GHA) & -\sin(GHA) & 0 \\ \sin(GHA) & \cos(GHA) & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

5. Transform the Earth-fixed coordinate axes to inertial:

$$\widehat{V}_N = [G]\widehat{N}$$
$$\widehat{V}_E = [G]\widehat{E}$$
$$\widehat{V}_V = [G]\widehat{V}$$

6. Compute the azimuth (AZ) and elevation (EL) of the Sun:

$$AZ = \arctan\left(\frac{\widehat{V}_N \cdot \widehat{R}_{Sun}}{\widehat{V}_E \cdot \widehat{R}_{Sun}}\right)$$

$$EL = \arcsin\left(\widehat{V}_V \cdot \widehat{R}_{Sun}\right)$$

E.3.3 Computational Complexity

The algorithm is of constant complexity ("O(C)"). All computations are double-precision floating point, and their cost will dominate. Table E–1 indicates the type and number of floating point operations required to compute the Sun azimuth and elevation of a single data. GCI Sun vector and GHA computation complexities are listed separately.

Assuming one computation for the scene center per scene 1227 + 20n (n = the mean number of floating point operations to compute a trigonometric function) floating point operations will be required for each scene. At 250 scenes per day, approximately 306,750 + 5,000n floating point operations per day will be required.

Table E–1. Floating Point Operations for WRS Scene Identifier Algorithms

Algorithm	+/–	–N	×	÷	MOD	Round	Trig
Azimuth/elevation	24	4	36	1	0	0	18
GCI Sun vector	7	0	7	0	1	0	5
GHA	579	4	560	5	1	1	45
Total	610	8	603	6	2	1	20

E.3.4 DSI Estimates

Table E–2 presents DSI estimates for each algorithm. For GCI Sun vector and GHA computations, the DSI estimate is a count of the DSI in the existing FORTRAN implementations. For the top-level algorithm, the estimate is based on the number of floating point operations required (assuming 1 DSI per operation); the number of distinct intermediate values computed (assuming 1 DSI per value); a 30-percent overhead for initialization and termination handling, iteration, etc.; and 100-percent overhead for exception handling.

Table E–2. DSI Estimates for WRS Scene Identification Algorithms

Algorithm	Estimated DSI
Sun azimuth/elevation at spacecraft	198
Sun azimuth/elevation at Earth	226
GCI Sun vector	22
GHA	218
Total	664

E.3.5 Sun Azimuth and Elevation Algorithm Issues and Questions

Time conversions required (e.g., from UTC to A.1) have not been incorporated into the algorithm.

Supplying the A.1-UT1 time difference to subroutine JGHAX, which computes the GHA, has not been incorporated into the algorithm. A direct transformation from UTC to UT1 may be more appropriate to the LPS version of this subroutine.

E.3.6 GCI Sun Vector Subroutine in FORTRAN

```
C          DATA SET SOL        AT LEVEL 001 AS OF 01/23/86

C**********************************************************************

C*

C*  NAME:

C*      SOL

C*

C*  PURPOSE:

C*      TO DETERMINE THE POSITION OF THE SUN WITH RESPECT TO C*      THE EARTH

C*

C*  COMPUTER:

C*      NAS R1/R2 MVS OPERATING SYSTEM

C*

C*  LANGUAGE:

C*      VS FORTRAN

C*

C*  METHOD:

C*      POSITION OF THE SUN WITH RESPECT TO THE EARTH IS

C*      DETERMINED FROM THE MEAN MOTION OF THE SUN AS

C*      DESCRIBED IN THE SUPPLEMENT TO THE NAUTICAL

C*      EPHEMERIS SOLAR POSITION AND VELOCITY ROUTINE

C*

C*      POSITIONAL ACCURACY ABOUT .0005 RADIAN OR 75000 KM

C*

C*     FORMULAE ARE FROM THE EXPLANATORY SUPPLEMENT TO C*THE

C*     ASTRONOMICAL EPHEMERIS AND THE AMERICAN EPHEMERIS AND NAUTICAL

C*     NAUTICAL ALMANAC (1961), PAGE 98.

C*

C*  REFERENCE :
```

```
C*     CSC/TM-77/6095, “SUBROUTINE DESCRIPTION AND MATHEMATICAL MODEL
C*     FOR THE OSMEAN MEAN ELEMENTS SUBROUTINE”, D.L RICHARDSON AND
C*     D. W. DUNHAM, APRIL, 1977, APPENDIX D.
C*
C*  CALLING SEQUENCE
C*     CALL SOL( AJD, X1,Y1, Z1, AL1)
C*
C*  ARGUMENT LIST
C*      NAME     TYPE   USE   DESCRIPTION
C*      ====     ====   ===   =============
C*      AJD      R*8    I     FULL JULIAN EPHEMERIS DATE
C*      X1       R*8    O     X-COMPONENT OF GEOCENTRIC SOLAR POSITION
C*                              VECTOR = X/R
C*      Y1       R*8    O     Y-COMPONENT OF GEOCENTRIC SOLAR POSITION
C*                              VECTOR = Y/R
C*      Z1       R*8    O     Z-COMPONENT OF GEOCENTRIC SOLAR POSITION
C*                              VECTOR = Z/R
C*      AL1      R*8    O     SCALING FACTOR FOR GEOCENTRIC SOLAR
C*                              DISTANCE
C*
C*  EXTERNAL VARIABLES:
C*      NAME     TYPE   USE   DESCRIPTION
C*      ====     ====   ===   ===============
C*    COMMON BLOCK:  CONSTA
C*      TWOPI    R*8     I    VALUE OF 2 * PI
C*      DEG2RD   R*8     I    DEGREES TO RADIANS CONVERSION FACTOR
C*
C*
C*  EXTERNAL REFERENCES:
```

```
C*      NONE
C*
C*  I/O PROCESSING:
C*      NONE
C*
C*  ERROR PROCESSING:
C*      NONE
C*
C*  RESTRICTIONS:
C*      NONE
C*
C*
C*  HISTORY
C*      NAME          DATE     COMMENTS
C*      =======       ======   ==========
C*      D. DUNHAM     12/76    ORIGINATOR
C*      B. JEFFE      4/85     UPDATED TO VSFORT, ADDED PROLOG
C*
C***********************************************************************
C*
      SUBROUTINE SOL(AJD, X1, Y1, Z1, AL1)
C
      IMPLICIT REAL*8 (A - H, O - Z)
C
      COMMON /CONSTA/RAD, PI, TWOPI, DEG2RD
      DATA ISTART /0/
C
C********* FOLLOWING ARE FOR BENCHMARK RUNS W/OLD CONSTANTS - REMOVE**
C
```

```
C     TWOPI = 6.2831853071796D0
C     DEG2RD = .17453292519943D-01
C
      IF (ISTART .EQ. 0) THEN
C
C  INITIALIZE CONSTANTS AND RESET ISTART VARIABLE ON 1ST ENTRY ONLY
C
           ISTART = 1
           GAM0 = 281.220833D0 * DEG2RD
           GAMD = .0000470684D0 * DEG2RD
           G0   = 358.475845D0 * DEG2RD
           GD   = 0.985600267D0 * DEG2RD
           EPS0 = 23.452294D0 * DEG2RD
           EPSD = 0.35626D-06 * DEG2RD
      ENDIF
C
      D   = AJD - 2415020.0D0                        ± E
      E   = 0.01675104D0 - 0.11444D-08*D        ± .1444E-8 * E
      E1  = 2.D0*E
      EPS = EPS0 - D*EPSD
      GAM = GAM0 + GAMD*D
      G   = G0 + D*GD
      AL  = GAM + G + E1*DSIN (G)
      AL  = DMOD (AL, TWOPI)
      X1  = DCOS (AL)
      Z1  = DSIN (AL)
      Y1  = Z1*DCOS (EPS)
      Z1  = Z1*DSIN (EPS)
      AL1 = E*DCOS (G)
```

```
C
C********* FOLLOWING RESETS TO OLD CONSTANTS IN COMMON - REMOVE****
C
C     TWOPI = 6.2831853D0
C     DEG2RD = .01745329252D0
C
      RETURN
      END
```

E.3.7 Greenwich Hour Angle Subroutine in FORTRAN

```
C/ ADD NAME=UTJGHAX                                04MAY90 15.38.35
C          DATA SET UTJGHAX    AT LEVEL 003 AS OF 10/30/89
C          DATA SET UTJGHAX    AT LEVEL 003 AS OF 01/31/89
C          DATA SET UTJGHAX    AT LEVEL 008 AS OF 11/16/88
C          DATA SET UTJGHAX    AT LEVEL 001 AS OF 07/21/88
C          DATA SET UTJGHAX    AT LEVEL 001 AS OF 06/20/88
       SUBROUTINE JGHAX(XJDA1,A1MUT1,GHA)
CC
CC PURPOSE: COMPUTES GREENWICH HOUR ANGLE USING ANALYTICAL DATA
CC
CC METHOD:  GREENWICH HOUR ANGLE IS COMPUTED AS THE SUM OF THE
CC          GREENWICH MEAN SIDEREAL TIME AT 0.0H OF DAY, THE
CC          UNIVERSAL TIME AND THE NUTATION CORRECTION TERM.
CC
CC ARGUMENT LIST:
CC   ARGUMENT  TYPE  I/O  DESCRIPTION <DIM>
CC     XJDA1    R8    I    A.1 JULIAN DATE
CC     A1MUT1   R8    I    A.1-UT1 DIFFERENCE (SEC)
CC     GHA      R8    O    GREENWICH HOUR ANGLE (RAD)
CC
```

```
CC CALLING SUBROUTINE: USER ROUTINE
CC
CC SUBROUTINES CALLED: JNUTON
CC
CC COMMON BLOCK VARIABLES USED: NONE
CC
CC MODIFICATIONS
CC NAME           DATE       DESCRIPTION
CC J. ROITZ       10/21/88   1) ADD A1-UT1 DIFFERENCE TO THE
CC                              ARGUMENT LIST AS UT1 IS NEEDED TO
CC                              COMPUTE GMST
CC                           2) CORRECTLY DETERMINE THE NUTATION
CC                              CORRECTION
CC
CC   -DECLARE EXTERNAL VARIABLES
       REAL*8 XJDA1,A1MUT1,GHA
CC   -DECLARE LOCAL VARIABLES
       REAL*8 C0,C1,C2,C3,TWOPI,ROTAT,GHA1,GHA2,GHA3,UT1,UTNOON,UTMID,
     $ ANUT(3,3), CNT, DLONG, TROBL, CTROBL
       DATA C0    /   1.7533685592332653D0    /
       DATA C1    / 628.33197068884084D0      /
       DATA C2    /   0.67707139449033354D-05 /
       DATA C3    /  -0.45087672343186841D-09 /
       DATA TWOPI /   6.2831853071795865D0    /
       DATA ROTAT /   7.2921151467D-05        /
CC
CC   -DETERMINE UT1
CC
       UT1 = XJDA1 - A1MUT1/86400.0D0
```

```
CC
CC   -COMPUTE NUMBER OF CENTURIES FROM J2000.0
CC
CC    FUNCTION DNINT ROUNDS TO THE NEAREST WHOLE NUMBER, THUS
CC    NOON OF THE CURRENT DAY.  SUBTRACTING HALF A DAY RESULTS IN
CC    0 HR OF THE CURRENT DAY.
CC
       UTNOON=DNINT(UT1)
       UTMID=UTNOON-0.5D0
       CNT=(UTMID-2451545.D0)/36525.D0
CC   -COMPUTE GREENWICH MEAN SIDEREAL ANGLE
       GHA1=C0+(C1+(C2+C3*CNT)*CNT)*CNT
       GHA2=ROTAT*86400.D0*(UT1-UTMID)
CC   -COMPUTE NUTATION CORRECTION
CC    THE NUTATION CORRECTION, GHA3, IS DEFINED AS THE PRODUCT OF
CC     COSINE OF TRUE OBLIQUITY AND NUTATION IN LONGITUDE.
CC     IN THE NUTATION MATRIX A,
CC     A(2,1) IS THE PRODUCT OF THE COSINE OF TRUE OBLIQUITY AND
CC            THE SINE OF THE NUTATION IN LONGITUDE.
CC     A(3,1) IS THE PRODUCT OF THE SINE OF TRUE OBLIQUITY AND
CC            THE SINE OF THE NUTATION IN LONGITUDE.
CC     A(3,1)/A(2,1) IS THE TANGENT OF THE TRUE OBLIQUITY.
CC     THESE WILL BE USED TO DETERMINE THE NUTATION IN LONGITUDE
CC     AND THEN GHA3
CC
       CALL JNUTON(XJDA1,ANUT,0,0)
       TROBL = DATAN2(ANUT(3,1),ANUT(2,1))
       CTROBL= DCOS(TROBL)
       DLONG = DASIN(ANUT(2,1)/CTROBL)
       GHA3 = DLONG*CTROBL
CC
CC   -ADD NUTATION CORRECTION TO MEAN VALUE AND SET POSITIVE < 2 PI
       GHA=DMOD(GHA1+GHA2+GHA3,TWOPI)
       IF(GHA.LT.0.D0)GHA=GHA+TWOPI
       RETURN
       END
C/ ADD NAME=UTJNUTON                               04MAY90 15.38.35
C          DATA SET UTJNUTON   AT LEVEL 003 AS OF 10/30/89
C          DATA SET UTJNUTON   AT LEVEL 002 AS OF 08/03/88
C          DATA SET UTJNUTON   AT LEVEL 001 AS OF 07/21/88
C          DATA SET UTJNUTON   AT LEVEL 001 AS OF 09/30/87
CCCC
C
      SUBROUTINE JNUTON (T2JA1, ANUT, IFDBG, IUDBG)
C
    1 FORMAT(‘ **** SUBROUTINE JNUTON  SEPTEMBER 15, 1987 ****’)
C
C-----LANGUAGE - VS FORTRAN
C
C-----FUNCTION - JNUTON COMPUTES THE NUTATION OF THE EARTH
C                AT TIME T2JA1 USING THE IAU 1980 THEORY OF NUTATION
C                EQUATIONS FOR THE MEAN-OF-J2000 COORDINATE SYSTEM.  THE
C                OUTPUT IS THE MEAN-EQUATOR-OF-DATE TO TRUE-EQUATOR-OF-
C                DATE ROTATION MATRIX.
C
C-----MATHEMATICAL METHOD - REFER TO THE SUPPLEMENT TO THE ASTRONOMICAL
C                           ALMANAC FOR 1984, PAGES S15 AND S23-S26.
C                           THIS SUBROUTINE IS A TRUNCATED VERSION OF
```

```
C                           THE NUTATION ALGORITHM. IT INCLUDES THE
C                           FIRST SECULAR TERM FOR EACH SERIES AND ALL
C                           TERMS WITH A COEFFICIENT OF 0.001 ARCSECOND
C                           OR GREATER IN MAGNITUDE (TERMS 1-4, 9-19,
C                           AND 31-50).
C                           THE CORRECT TIME INPUT TO THE ALGORITHM IS
C                           BARYCENTRIC DYNAMICAL TIME (TDB). THE TDT
C                           (OR EPHEMERIS TIME) TO TDB CORRECTION IS
C                           PERIODIC  WITH THE AMPLITUDE LESS THAN
C                           0.002 SECONDS.
C                           THE THREE ROTATION ANGLES ARE:
C                            1) MEAN OBLIQUITY OF THE ECLIPTIC
C                               (MEAN-EQUATOR- TO MEAN-ECLIPTIC-OF-DATE)
C                            2) NEGATIVE NUTATION IN LONGITUDE
C                            3) NEGATIVE TRUE OBLIQUITY OF THE ECLIPTIC
C                               (TRUE-ECLIPTIC- TO TRUE-EQUATOR-OF-DATE)
C
C-----ARGUMENTS -
C     ARGUMENT  TYPE  IO  DESCRIPTION
C     --------  ----  --  -----------
C     T2JA1     R*8   I  A.1 JULIAN DATE OF THE EPOCH IN DAYS PLUS
C                         FRACTION OF DAY
C     ANUT(3,3) R*8   O  ROTATION MATRIX FOR NUTATION FOR T2JA1
C                         (MEAN-EQUATOR-OF-DATE TO TRUE-EQUATOR-OF-DATE)
C     IFDBG     I*4   I  FLAG TO REQUEST DEBUG PRINTER OUTPUT
C                                 = 0 ... NO PRINTER OUTPUT
C                                 = 1 ... PRINT INPUT AND OUTPUT
C                                 = 2 ... PRINT INPUT, OUTPUT, CONSTANTS
C     IUDBG     I*4   I  FORTRAN UNIT NUMBER FOR DEBUG PRINTER OUTPUT
```

```
C
C-----EXTERNAL REFERENCES -
C     JOBLTY - COMPUTES THE MEAN OBLIQUITY OR TRUE OBLIQUITY AND THE
C              ECLIPTIC TO EQUATOR ROTATION MATRIX
C
C-----CALLED BY - JPRENU OR USER ROUTINE
C
C-----COMMONS REFERENCED - JCOEF
C-----VARIABLES -
C     VARIABLE  TYPE      DESCRIPTION
C     --------  ----      -----------
C     TJDSTD    R*8      STANDARD EPOCH FOR THE J2000 COORDINATE SYSTEM
C                         IN DAYS PLUS FRACTION OF DAY
C     DAYCEN    R*8      NUMBER OF DAYS IN A JULIAN CENTURY
C     DA1TDT    R*8      A.1 ATOMIC TIME TO TERRESTRIAL DYNAMICAL TIME
C                         (TDT) DIFFERENCE IN DAYS
C     ARCTR     R*8      ARCSECONDS TO RADIANS CONVERSION FACTOR
C     CFARGS    R*8      COEFFICIENTS FOR COMPUTING THE FUNDAMENTAL
C                         ARGUMENTS OF THE NUTATION SERIES IN ARCSECONDS
C     CFMULT    R*8      COEFFICIENTS OF THE FUNDAMENTAL ARGUMENTS FOR
C                         THE NUTATION SERIES IN ARCSECONDS
C     CFDLON    R*8      COEFFICIENTS OF THE LONGITUDE NUTATION TERMS
C                         FOR THE NUTATION SERIES IN ARCSECONDS
C     CFTDLN    R*8      COEFFICIENTS OF THE SECULAR LONGITUDE NUTATION
C                         TERMS FOR THE NUTATION SERIES IN ARCSECONDS
C     NTERMS    I*4      THE NUMBER OF NONSECULAR TERMS INCLUDED IN THE
C                         COMPUTATION
C     NSECL     I*4      THE NUMBER OF SECULAR TERMS INCLUDED IN THE
C                         COMPUTATION
```

```
C
C-----ERROR HANDLING - NONE
C
C-----FILES REFERENCED - NONE
C
C-----DESIGNER - S. DEVLIN, CSC, AUGUST 20, 1987
C
C-----PROGRAMMER - S. DEVLIN, CSC  SEPTEMBER 15, 1987
C
C-----VERIFIED BY -
C
C-----MODIFICATIONS -
C     NAME   DATE          DESCRIPTION
C
CCCC
C    -DECLARE ARGUMENTS
      REAL*8 T2JA1
      REAL*8 ANUT(3,3)
      INTEGER*4 IFDBG
      INTEGER*4 IUDBG
C    -DECLARE LOCAL VARIABLES
      REAL*8 TREF
      REAL*8 ARGS(5)
      REAL*8 ANGL, DLONG
      REAL*8 AMOB(3,3), ATOB(3,3)
      REAL*8 COBLTM, SOBLTM, CTROBL, STROBL, CDLONG, SDLONG
      INTEGER*4 I, J, IOPT
C
C    -COMMON BLOCK JCOEF VARIABLES
```

```
      REAL*8 TJDSTD
      REAL*8 DAYCEN, DA1TDT, ARCTR
      REAL*8 CFZETA(6), CFZEE(6), CFTHET(6)
      REAL*8 CFOBLM(4)
      REAL*8 CFARGS(4,5)
      REAL*8 CFMULT(5,35), CFDLON(35), CFDOBL(35)
      REAL*8 CFTDLN(1), CFTDOB(1)
      INTEGER*4 NTERMS
      INTEGER*4 NSECL
C
      COMMON /JCOEF/ TJDSTD, DAYCEN, DA1TDT, ARCTR,
     2               CFZETA, CFZEE, CFTHET,
     3               CFOBLM, CFARGS, CFMULT, CFDLON, CFDOBL,
     4               CFTDLN, CFTDOB, NTERMS, NSECL
C
C
C     PRINT DEBUG OUTPUT
      IF(IFDBG.GT.0) THEN
        WRITE(IUDBG,1)
        WRITE(IUDBG,1000) T2JA1
        IF(IFDBG.GT.1) THEN
          WRITE(IUDBG,2003) TJDSTD, DAYCEN, DA1TDT, ARCTR
          WRITE(IUDBG,2004) NTERMS, NSECL, CFTDLN
          WRITE(IUDBG,2006) (I,(CFMULT(J,I),J=1,5),CFDLON(I),I=1,NTERMS)
        END IF
      END IF
C
C     COMPUTE THE TIME FROM J2000 TO THE EPOCH TIME IN CENTURIES
      TREF = ((T2JA1+DA1TDT) - TJDSTD) / DAYCEN
```

```
C
C     COMPUTE FUNDAMENTAL ARGUMENTS FOR NUTATION SERIES
C
      DO 100 I=1,5
        ARGS(I) = ARCTR * (CFARGS(1,I) + TREF * (CFARGS(2,I) +
     2             TREF * (CFARGS(3,I) + TREF * CFARGS(4,I)) ))
  100 CONTINUE
C
C     COMPUTE NUTATION IN LONGITUDE
C
      DLONG = 0.0D0
      DO 300 I=1,NTERMS
C
        ANGL = 0.0D0
        DO 200 J=1,5
          ANGL = ANGL + CFMULT(J,I) * ARGS(J)
  200   CONTINUE
C
        IF(I.GT.NSECL) THEN
          DLONG = DLONG + CFDLON(I) * DSIN(ANGL)
        ELSE
          DLONG = DLONG + (CFDLON(I) + CFTDLN(I)*TREF) * DSIN(ANGL)
        END IF
  300 CONTINUE
      DLONG = ARCTR * DLONG
C
C     COMPUTE MEAN OBLIQUITY AND TRUE OBLIQUITY
C
      IOPT = 1
      CALL JOBLTY(T2JA1, AMOB, IOPT, IFDBG, IUDBG)
      IOPT = 2
      CALL JOBLTY(T2JA1, ATOB, IOPT, IFDBG, IUDBG)
C
C     COMPUTE THE ROTATION MATRIX
C
      CDLONG = DCOS(DLONG)
      SDLONG = DSIN(DLONG)
      COBLTM = AMOB(2,2)
      SOBLTM = AMOB(3,2)
      CTROBL = ATOB(2,2)
      STROBL = ATOB(3,2)
C
      ANUT(1,1) =  CDLONG
      ANUT(1,2) = -SDLONG*COBLTM
      ANUT(1,3) = -SDLONG*SOBLTM
      ANUT(2,1) =  CTROBL*SDLONG
      ANUT(2,2) =  CTROBL*CDLONG*COBLTM + STROBL*SOBLTM
      ANUT(2,3) =  CTROBL*CDLONG*SOBLTM - STROBL*COBLTM
      ANUT(3,1) =  STROBL*SDLONG
      ANUT(3,2) =  STROBL*CDLONG*COBLTM - CTROBL*SOBLTM
      ANUT(3,3) =  STROBL*CDLONG*SOBLTM + CTROBL*COBLTM
C
C     PRINT DEBUG OUTPUT
      IF(IFDBG.GT.0) THEN
        WRITE(IUDBG,2001) TREF, DLONG, ARGS
        WRITE(IUDBG,2002) ((ANUT(I,J),J=1,3),I=1,3)
      END IF
C
```

```
 1000 FORMAT(‘ T2JA1 = ‘,F18.8)
 2001 FORMAT(‘0TREF, DLONG = ‘,2D24.15,/,
     $ ‘ ARGS(1)= ‘,D22.14,/,
     $ ‘ ARGS(2)= ‘,D22.14,/,
     $ ‘ ARGS(3)= ‘,D22.14,/,
     $ ‘ ARGS(4)= ‘,D22.14,/,
     $ ‘ ARGS(5)= ‘,D22.14)
 2002 FORMAT(‘ ANUT(1,1-3) = ‘,3F20.15,/’ ANUT(2,1-3) = ‘,3F20.15,
     2  /’ ANUT(3,1-3) = ‘,3F20.15)
 2003 FORMAT(‘ TJDSTD, DAYCEN, DA1TDT= ‘,F18.8,F10.2,D24.15)
 2004 FORMAT(‘ NTERMS, NSECL = ‘,2I4,
     $ ‘ CFTDLN(1) = ‘,D22.12,/,
     $ ‘ CFTDLN(2) = ‘,D22.12,/,
     $ ‘ CFTDLN(3) = ‘,D22.12,/,
     $ ‘ CFTDLN(4) = ‘,D22.12)
 2006 FORMAT(1X,I4,5F5.0,D22.12)
C
      RETURN
      END
C/ ADD NAME=UTJOBLTY                               04MAY90 15.38.35
C          DATA SET UTJOBLTY   AT LEVEL 003 AS OF 10/30/89
C          DATA SET UTJOBLTY   AT LEVEL 002 AS OF 08/03/88
C          DATA SET UTJOBLTY   AT LEVEL 001 AS OF 07/21/88
C          DATA SET UTJOBLTY   AT LEVEL 001 AS OF 09/30/87
CCCC
C
      SUBROUTINE JOBLTY (T2JA1, AOBL, IOPT, IFDBG, IUDBG)
C
    1 FORMAT(‘ **** SUBROUTINE JOBLTY  SEPTEMBER 15, 1987 ****’)
```

```
C
C-----LANGUAGE - VS FORTRAN
C
C-----FUNCTION - JOBLTY COMPUTES THE MEAN OBLIQUITY OF THE EQUATOR OR
C                THE TRUE OBLIQUITY OF THE EQUATOR AT TIME T2JA1 USING
C                THE IAU 1980 THEORY OF NUTATION EQUATIONS FOR THE
C                MEAN-OF-J2000 COORDINATE SYSTEM.  THE OUTPUT IS THE
C                ECLIPTIC-OF-DATE TO EQUATOR-OF-DATE ROTATION MATRIX.
C
C-----MATHEMATICAL METHOD - REFER TO THE SUPPLEMENT TO THE ASTRONOMICAL
C                           ALMANAC FOR 1984, PAGES S15 AND S23-S26.
C                           FOR THE TRUE-OF-DATE COORDINATE SYSTEM, THE
C                           COMPUTATION IS A TRUNCATED VERSION OF
C                           THE NUTATION ALGORITHM. IT INCLUDES THE
C                           FIRST SECULAR TERM FOR THE SERIES AND ALL
C                           TERMS WITH A COEFFICIENT OF 0.001 ARCSECOND
C                           OR GREATER IN MAGNITUDE (TERMS 1-4, 9-19,
C                           AND 31-50).
C                           THE CORRECT TIME INPUT TO THE ALGORITHM IS
C                           BARYCENTRIC DYNAMICAL TIME (TDB). THE TDT
C                           (OR EPHEMERIS TIME) TO TDB CORRECTION IS
C                           PERIODIC  WITH THE AMPLITUDE LESS THAN
C                           0.002 SECONDS.
C
C-----ARGUMENTS -
C     ARGUMENT  TYPE  IO  DESCRIPTION
C     --------  ----  --  -----------
C     T2JA1     R*8   I  A.1 JULIAN DATE OF THE EPOCH IN DAYS PLUS
C                         FRACTION OF DAY
```

```
C     AOBL(3,3) R*8   O  ROTATION MATRIX FOR OBLIQUITY FOR T2JA1
C     IOPT      I*4   I  FLAG TO INDICATE THE DESIRED ROTATION
C                                 = 1 ... MEAN ECLIPTIC OF DATE TO MEAN
C                                     EQUATOR OF DATE (MEAN OBLIQUITY)
C                                 = 2 ... TRUE ECLIPTIC OF DATE TO TRUE
C                                     EQUATOR OF DATE (TRUE OBLIQUITY)
C     IFDBG     I*4   I  FLAG TO REQUEST DEBUG PRINTER OUTPUT
C                                 = 0 ... NO PRINTER OUTPUT
C                                 = 1 ... PRINT INPUT AND OUTPUT
C                                 = 2 ... PRINT INPUT, OUTPUT, CONSTANTS
C     IUDBG     I*4   I  FORTRAN UNIT NUMBER FOR DEBUG PRINTER OUTPUT
C
C-----EXTERNAL REFERENCES - NONE
C
C-----CALLED BY - JNUTON, USER ROUTINE
C
C-----COMMONS REFERENCED - JCOEF
C-----VARIABLES -
C     VARIABLE  TYPE      DESCRIPTION
C     --------  ----      -----------
C     TJDSTD    R*8      STANDARD EPOCH FOR THE J2000 COORDINATE SYSTEM
C                         IN DAYS PLUS FRACTION OF DAY
C     DAYCEN    R*8      NUMBER OF DAYS IN A JULIAN CENTURY
C     DA1TDT    R*8      A.1 ATOMIC TIME TO TERRESTRIAL DYNAMICAL TIME
C                         (TDT) DIFFERENCE IN DAYS
C     ARCTR     R*8      ARCSECONDS TO RADIANS CONVERSION FACTOR
C     CFOBLM    R*8      COEFFICIENTS FOR THE MEAN OBLIQUITY COMPUTATION
C                         IN ARCSECONDS
C     CFARGS    R*8      COEFFICIENTS FOR COMPUTING THE FUNDAMENTAL
```

```
C                          ARGUMENTS OF THE NUTATION SERIES IN ARCSECONDS
C     CFMULT    R*8      COEFFICIENTS OF THE FUNDAMENTAL ARGUMENTS FOR
C                          THE NUTATION SERIES IN ARCSECONDS
C     CFDOBL    R*8      COEFFICIENTS OF THE OBLIQUITY NUTATION TERMS
C                          FOR THE NUTATION SERIES IN ARCSECONDS
C     CFTDOB    R*8      COEFFICIENTS OF THE SECULAR OBLIQUITY NUTATION
C                          TERMS FOR THE NUTATION SERIES IN ARCSECONDS
C     NTERMS    I*4      THE NUMBER OF NONSECULAR TERMS INCLUDED IN THE
C                          COMPUTATION
C     NSECL     I*4      THE NUMBER OF SECULAR TERMS INCLUDED IN THE
C                          COMPUTATION
C
C-----ERROR HANDLING - NONE
C
C-----FILES REFERENCED - NONE
C
C-----DESIGNER - S. DEVLIN, CSC, SEPTEMBER 14, 1987
C
C-----PROGRAMMER - S. DEVLIN, CSC  SEPTEMBER 15, 1987
C
C-----VERIFIED BY -
C
C-----MODIFICATIONS -
C     NAME   DATE          DESCRIPTION
C
CCCC
C    -DECLARE ARGUMENTS
      REAL*8 T2JA1
      REAL*8 AOBL(3,3)
      INTEGER*4 IFDBG
      INTEGER*4 IUDBG
C    -DECLARE LOCAL VARIABLES
      REAL*8 TREF
      REAL*8 ARGS(5)
      REAL*8 ANGL, DOBLT, OBLTMR, OBLTYR
      REAL*8 COBLTY, SOBLTY
      INTEGER*4 I, J, IOPT
C
C    -COMMON BLOCK JCOEF VARIABLES
      REAL*8 TJDSTD
      REAL*8 DAYCEN, DA1TDT, ARCTR
      REAL*8 CFZETA(6), CFZEE(6), CFTHET(6)
      REAL*8 CFOBLM(4)
      REAL*8 CFARGS(4,5)
      REAL*8 CFMULT(5,35), CFDLON(35), CFDOBL(35)
      REAL*8 CFTDLN(1), CFTDOB(1)
      INTEGER*4 NTERMS
      INTEGER*4 NSECL
C
      COMMON /JCOEF/ TJDSTD, DAYCEN, DA1TDT, ARCTR,
     2               CFZETA, CFZEE, CFTHET,
     3               CFOBLM, CFARGS, CFMULT, CFDLON, CFDOBL,
     4               CFTDLN, CFTDOB, NTERMS, NSECL
C
C
C     PRINT DEBUG OUTPUT
      IF(IFDBG.GT.0) THEN
        WRITE(IUDBG,1)
        WRITE(IUDBG,1000) T2JA1, IOPT
        IF(IFDBG.GT.1) THEN
          WRITE(IUDBG,2003) TJDSTD, DAYCEN, DA1TDT, ARCTR
        END IF
      END IF
C
C     COMPUTE THE TIME FROM J2000 TO THE EPOCH TIME IN CENTURIES
      TREF = ((T2JA1+DA1TDT) - TJDSTD) / DAYCEN
C
C     COMPUTE MEAN OBLIQUITY
C
      OBLTMR = ARCTR * (CFOBLM(1) + TREF * (CFOBLM(2) +
     2             TREF * (CFOBLM(3) + TREF * CFOBLM(4)) ))
C
      IF(IOPT.EQ.1) THEN
C       MEAN-OF-DATE
        OBLTYR = OBLTMR
      ELSE
C
C       TRUE-OF-DATE
C       COMPUTE FUNDAMENTAL ARGUMENTS FOR NUTATION SERIES
C
        DO 100 I=1,5
          ARGS(I) = ARCTR * (CFARGS(1,I) + TREF * (CFARGS(2,I) +
     2               TREF * (CFARGS(3,I) + TREF * CFARGS(4,I)) ))
  100   CONTINUE
C
C       COMPUTE NUTATION IN OBLIQUITY
C
```

```
        DOBLT = 0.0D0
        DO 300 I=1,NTERMS
C
          ANGL = 0.0D0
          DO 200 J=1,5
            ANGL = ANGL + CFMULT(J,I) * ARGS(J)
  200     CONTINUE
C
          IF(I.GT.NSECL) THEN
            DOBLT = DOBLT + CFDOBL(I) * DCOS(ANGL)
          ELSE
            DOBLT = DOBLT + (CFDOBL(I) + CFTDOB(I)*TREF) * DCOS(ANGL)
          END IF
  300   CONTINUE
        DOBLT = ARCTR * DOBLT
C
        OBLTYR = OBLTMR + DOBLT
      END IF
C
C     COMPUTE THE ROTATION MATRIX
C
      COBLTY = DCOS(OBLTYR)
      SOBLTY = DSIN(OBLTYR)
C
      AOBL(1,1) =  1.0D0
      AOBL(1,2) =  0.0D0
      AOBL(1,3) =  0.0D0
      AOBL(2,1) =  0.0D0
      AOBL(2,2) =  COBLTY
      AOBL(2,3) = -SOBLTY
      AOBL(3,1) =  0.0D0
      AOBL(3,2) =  SOBLTY
      AOBL(3,3) =  COBLTY
C
C     PRINT DEBUG OUTPUT
      IF(IFDBG.GT.0) THEN
        WRITE(IUDBG,2001) TREF, OBLTMR, DOBLT, OBLTYR, ARGS
        WRITE(IUDBG,2002) ((AOBL(I,J),J=1,3),I=1,3)
      END IF
C
 1000 FORMAT(‘ T2JA1 = ‘,F18.8,’ IOPT = ‘,I4)
 2001 FORMAT(
     $       ‘ TREF    = ‘,D24.15,/,
     $       ‘ OBLTMR  = ‘,D24.15,/,
     $       ‘ DOBLT   = ‘,D24.15,/,
     $       ‘ OBLTYR  = ‘,D24.15,/,
     $       ‘ ARGS(1) = ‘,D22.14,/,
     $       ‘ ARGS(2) = ‘,D22.14,/,
     $       ‘ ARGS(3) = ‘,D22.14,/,
     $       ‘ ARGS(4) = ‘,D22.14,/,
     $       ‘ ARGS(5) = ‘,D22.14)
 2002 FORMAT(‘ AOBL(1,1-3) = ‘,3F20.15,/’ AOBL(2,1-3) = ‘,3F20.15,
     2  /’ AOBL(3,1-3) = ‘,3F20.15)
 2003 FORMAT(‘ TJDSTD, DAYCEN, DA1TDT = ‘,F18.8,F10.2,D24.15)
C
      RETURN
      END
C/ ADD NAME=UTBJCOEF                               04MAY90 15.38.35
```

```
C          DATA SET UTBJCOEF   AT LEVEL 003 AS OF 10/30/89
C          DATA SET UTBJCOEF   AT LEVEL 002 AS OF 08/03/88
C          DATA SET UTBJCOEF   AT LEVEL 001 AS OF 07/21/88
C          DATA SET UTBJCOEF   AT LEVEL 001 AS OF 10/16/87
C          DATA SET BDJCOEF    AT LEVEL 001 AS OF 10/12/87
CCCC
C
      BLOCKDATA
C
CX  1 FORMAT(‘ **** BLOCKDATA BDJCOEF  OCTOBER 10, 1987 ****’)
C
C-----LANGUAGE - VS FORTRAN
C
C-----FUNCTION - COMMON BLOCK JCOEF CONTAINS COEFFICIENTS FOR THE
C                PRECESSION COMPUTATION FOR THE MEAN-OF-J2000 COORDINATE
C                SYSTEM AND COEFFICIENTS FOR THE TRUNCATED VERSION OF
C                THE NUTATION SERIES COMPUTATION USING THE IAU 1980
C                THEORY OF NUTATION EQUATIONS FOR THE MEAN-OF-J2000
C                COORDINATE SYSTEM.
C
C
C-----VARIABLES -
C     VARIABLE  TYPE      DESCRIPTION
C     --------  ----      -----------
C     TJDSTD    R*8      STANDARD EPOCH FOR THE J2000 COORDINATE SYSTEM
C                         IN DAYS PLUS FRACTION OF DAY
C     DAYCEN    R*8      NUMBER OF DAYS IN A JULIAN CENTURY
C     DA1TDT    R*8      A.1 ATOMIC TIME TO TERRESTRIAL DYNAMICAL TIME
C                         (TDT) DIFFERENCE IN DAYS
```

```
C                         ( (32.184D0-0.0343817D0) / 86400.0D0 )
C     ARCTR     R*8      ARCSECONDS TO RADIANS CONVERSION FACTOR
C                         (0.0174532925199433D0 / 3600.D0)
C     CFZETA    R*8      COEFFICIENTS FOR PRECESSION ANGLE ZETA
C                         IN ARCSECONDS
C     CFZEE     R*8      COEFFICIENTS FOR PRECESSION ANGLE Z
C                         IN ARCSECONDS
C     CFTHET    R*8      COEFFICIENTS FOR PRECESSION ANGLE THETA
C                         IN ARCSECONDS
C     CFOBLM    R*8      COEFFICIENTS FOR THE MEAN OBLIQUITY COMPUTATION
C                         IN ARCSECONDS
C     CFARGS    R*8      COEFFICIENTS FOR COMPUTING THE FUNDAMENTAL
C                         ARGUMENTS OF THE NUTATION SERIES IN ARCSECONDS
C     CFMULT    R*8      COEFFICIENTS OF THE FUNDAMENTAL ARGUMENTS FOR
C                         THE NUTATION SERIES IN ARCSECONDS
C     CFDLON    R*8      COEFFICIENTS OF THE LONGITUDE NUTATION TERMS
C                         FOR THE NUTATION SERIES IN ARCSECONDS
C     CFDOBL    R*8      COEFFICIENTS OF THE OBLIQUITY NUTATION TERMS
C                         FOR THE NUTATION SERIES IN ARCSECONDS
C     CFTDLN    R*8      COEFFICIENTS OF THE SECULAR LONGITUDE NUTATION
C                         TERMS FOR THE NUTATION SERIES IN ARCSECONDS
C     CFTDOB    R*8      COEFFICIENTS OF THE SECULAR OBLIQUITY NUTATION
C                         TERMS FOR THE NUTATION SERIES IN ARCSECONDS
C     NTERMS    I*4      THE NUMBER OF NONSECULAR TERMS INCLUDED IN THE
C                         COMPUTATION
C     NSECL     I*4      THE NUMBER OF SECULAR TERMS INCLUDED IN THE
C                         COMPUTATION
C
C-----DESIGNER - S. DEVLIN, CSC, SEPTEMBER 14, 1987
```

```
C
C-----PROGRAMMER - S. DEVLIN, CSC, SEPTEMBER 15, 1987
C
C-----VERIFIED BY -
C
C-----MODIFICATIONS -
C     NAME         DATE    DESCRIPTION
C     S. DEVLIN  10/12/87  ADD D0 TO CFMULT ARRAY VALUES
C
CCCC
C    -COMMON BLOCK JCOEF VARIABLES
      REAL*8 TJDSTD
      REAL*8 DAYCEN, DA1TDT, ARCTR
      REAL*8 CFZETA(6), CFZEE(6), CFTHET(6)
      REAL*8 CFOBLM(4)
      REAL*8 CFARGS(4,5)
      REAL*8 CFMULT(5,35), CFDLON(35), CFDOBL(35)
      REAL*8 CFTDLN(1), CFTDOB(1)
      INTEGER*4 NTERMS
      INTEGER*4 NSECL
C
      COMMON /JCOEF/ TJDSTD, DAYCEN, DA1TDT, ARCTR,
     2               CFZETA, CFZEE, CFTHET,
     3               CFOBLM, CFARGS, CFMULT, CFDLON, CFDOBL,
     4               CFTDLN, CFTDOB, NTERMS, NSECL
C
      DATA TJDSTD/2451545.0D0/
      DATA DAYCEN/36525.0D0/
      DATA DA1TDT/0.372102063657407D-03/
      DATA ARCTR/0.484813681109536D-05/
      DATA CFZETA /2306.2181D0,   1.39656D0, -0.000139D0,
     2               0.30188D0, -0.000344D0,  0.017998D0/
      DATA CFZEE  /2306.2181D0,   1.39656D0, -0.000139D0,
     2               1.09468D0,  0.000066D0,  0.018203D0/
      DATA CFTHET /2004.3109D0,  -0.85330D0, -0.000217D0,
     2              -0.42665D0, -0.000217D0, -0.041833D0/
      DATA CFOBLM/84381.448D0, -46.8150D0, -0.00059D0, 0.001813D0/
      DATA CFARGS/ 485866.733D0, 1717915922.633D0,  31.310D0,  0.064D0,
     2             1287099.804D0,  129596581.224D0,  -0.577D0, -0.012D0,
     3              335778.877D0, 1739527263.137D0, -13.257D0,  0.011D0,
     4             1072261.307D0, 1602961601.328D0,  -6.891D0,  0.019D0,
     4              450160.280D0,   -6962890.539D0,   7.455D0,  0.008D0/
      DATA CFMULT/0.D0,0.D0,0.D0,0.D0,1.D0,  0.D0,0.D0,0.D0,0.D0,2.D0,
     2            -2.D0,0.D0,2.D0,0.D0,1.D0,  2.D0,0.D0,-2.D0,0.D0,0.D0,
     3             0.D0,0.D0,2.D0,-2.D0,2.D0, 0.D0,1.D0,0.D0,0.D0,0.D0,
     4             0.D0,1.D0,2.D0,-2.D0,2.D0, 0.D0,-1.D0,2.D0,-2.D0,2.D0,
     5             0.D0,0.D0,2.D0,-2.D0,1.D0, 2.D0,0.D0,0.D0,-2.D0,0.D0,
     6             0.D0,0.D0,2.D0,-2.D0,0.D0, 0.D0,2.D0,0.D0,0.D0,0.D0,
     7             0.D0,1.D0,0.D0,0.D0,1.D0,  0.D0,2.D0,2.D0,-2.D0,2.D0,
     8             0.D0,-1.D0,0.D0,0.D0,1.D0, 0.D0,0.D0,2.D0,0.D0,2.D0,
     9             1.D0,0.D0,0.D0,0.D0,0.D0,  0.D0,0.D0,2.D0,0.D0,1.D0,
     A             1.D0,0.D0,2.D0,0.D0,2.D0,  1.D0,0.D0,0.D0,-2.D0,0.D0,
     B            -1.D0,0.D0,2.D0,0.D0,2.D0,  0.D0,0.D0,0.D0,2.D0,0.D0,
     C             1.D0,0.D0,0.D0,0.D0,1.D0, -1.D0,0.D0,0.D0,0.D0,1.D0,
     D            -1.D0,0.D0,2.D0,2.D0,2.D0,  1.D0,0.D0,2.D0,0.D0,1.D0,
     E             0.D0,0.D0,2.D0,2.D0,2.D0,  2.D0,0.D0,0.D0,0.D0,0.D0,
     F             1.D0,0.D0,2.D0,-2.D0,2.D0, 2.D0,0.D0,2.D0,0.D0,2.D0,
     G             0.D0,0.D0,2.D0,0.D0,0.D0, -1.D0,0.D0,2.D0,0.D0,1.D0,
     H           -1.D0,0.D0,0.D0,2.D0,1.D0,  1.D0,0.D0,0.D0,-2.D0,1.D0,
     I           -1.D0,0.D0,2.D0,2.D0,1.D0/
      DATA CFDLON/-17.1996D0,  0.2062D0, 0.0046D0,  0.0011D0, -1.3187D0,
     2              0.1426D0, -0.0517D0, 0.0217D0,  0.0129D0,  0.0048D0,
     3             -0.0022D0,  0.0017D0,-0.0015D0, -0.0016D0, -0.0012D0,
     4             -0.2274D0,  0.0712D0,-0.0386D0, -0.0301D0, -0.0158D0,
     5              0.0123D0,  0.0063D0, 0.0063D0, -0.0058D0, -0.0059D0,
     6             -0.0051D0, -0.0038D0, 0.0029D0,  0.0029D0, -0.0031D0,
     7              0.0026D0,  0.0021D0, 0.0016D0, -0.0013D0, -0.0010D0/
      DATA CFDOBL/9.2025D0, -0.0895D0, -0.0024D0,  0.D0,      0.5736D0,
     2            0.0054D0,  0.0224D0, -0.0095D0, -0.0070D0,  0.0001D0,
     3            0.0D0,     0.0D0,     0.0009D0,  0.0007D0,  0.0006D0,
     4            0.0977D0, -0.0007D0,  0.0200D0,  0.0129D0, -0.0001D0,
     5           -0.0053D0, -0.0002D0, -0.0033D0,  0.0032D0,  0.0026D0,
     6            0.0027D0,  0.0016D0, -0.0001D0, -0.0012D0,  0.0013D0,
     7           -0.0001D0, -0.0010D0, -0.0008D0,  0.0007D0,  0.0005D0/
      DATA CFTDLN/-0.01742D0/
      DATA CFTDOB/0.00089D0/
      DATA NTERMS/35/
      DATA NSECL/1/
C
      END
```

E.4 WRS Scene Center and Corner Algorithms

E.4.1 ETM+ Look Point Latitude and Longitude Computation

The spacecraft time, position, and attitude are used to compute the look point on the ground of the ETM+ line of sight to the scene center and each of the scene corners. Given the spacecraft time (*t*), ephemeris ($\vec{R}$ – GCI spacecraft position vector), attitude quaternion (q_1, q_2, q_3, q_4 – the Euler parameters), Earth semi-major axis (*a*) and semi-minor axis (*b*), transformation from ETM+ to spacecraft body matrix ($[T_{BM}]$), and ETM+ line-of-sight vector at the center of the mirror scan ($\vec{V}$), compute the geodetic latitude and longitude of the ground point in the view of ETM+. The following computations are performed for each ephemeris and attitude point:

1. Compute the attitude matrix (transformation from GCI to body) from the Euler parameters (quaternion) obtained from PCD:

$$[Q] = \begin{vmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2q_1q_3 - q_2q_4 \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{vmatrix}$$

   This computation is available in the Reusable Software Library (RSL) routine QTOA.

2. Compute the transformation from spacecraft body to GCI (inverse of $[Q]$):

$$[T_{IB}] = [Q]^{-1}$$

3. Compute the transformation from ETM+ to GCI:

$$[T_{IM}] = [T_{IB}][T_{BM}]$$

4. The assumption is that the scan associated with the current time point could contain the center of a scene or the corners. The scene center will be in the direction of the ETM+ line-of-sight vector at the center of the mirror scan (along the optical axis). The corners will be at either extreme of the mirror scan (the maximum along-scan angle) and offset from the center line of the scan by the maximum cross-scan angle. The cross-scan angle represents the half width of the detector array. The scan that contains the two scene corners that occur in time before the scene center will be referred to as the trailing edge scan; the scan containing the two corners that occur after the scene center will be referred to as the leading edge scan (the leading edge scan is the final scan taken for the current WRS scene).

   The cross-scan angle (σ) and along-scan angle (θ) for each corner are shown in the following tabulation. For the scene center, set $\sigma = \theta = 0$.

Trailing edge	θ	-7.695 deg	+7.695 deg
	σ	-0.01948 deg	-0.01948 deg
Leading edge	θ	-7.695 deg	+7.695 deg
	σ	+0.01948 deg	+0.01948 deg

The 7.695-degree value represents half of the nominal active scan length (from the optical axis to either scan end). The angular width seen by the detectors is specified as 680 micro radians; half of this is 340 micro radians or 0.01948 deg.

5. Compute a line-of-sight vector ($\hat{P}$) from the along-scan and cross-scan angles (it is necessary to compute only once using positive values of θ and σ):

$$r\begin{bmatrix}\cos(\phi)\cos(\lambda)\\ \cos(\phi)\sin(\lambda)\\ \sin(\phi)\end{bmatrix} = \vec{X} + d\left[T_{IM}\right]\begin{bmatrix}-\sin(\sigma)\\ -\cos(\sigma)\sin(\theta)\\ \cos(\sigma)\cos(\theta)\end{bmatrix}$$

Set the signs of the line-of-sight vector components for each corner as in the following tabulation:

	Trailing Edge		Leading Edge	
	Left	Right	Left	Right
$\hat{P}_X$	–	–	+	+
$\hat{P}_Y$	–	+	–	+
$\hat{P}_Z$	+	+	+	+

If $\sigma = \theta = 0$ (scene center), set $\hat{P} = \hat{V}$. $\hat{V}$ is nominally directed along the z axis of the ETM+ coordinate system, that is

$$\hat{V} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

6. Rotate $\hat{P}$ to vector $\hat{P}$ in GCI coordinates:

$$\hat{P} = \left[T_{IM}\right]\hat{P}$$

7. Compute the spacecraft geocentric latitude (ϕ) and right ascension (λ) from the position vector components:

$$\phi = \sin^{-1}\left(\frac{R_z}{R}\right)$$

$$\lambda = \tan^{-1}\left(\frac{R_y}{R_x}\right)$$

where $R = |\vec{R}|$, the magnitude of the spacecraft GCI position.

8. Compute a quadratic equation of the form $Ad^2 + 2Bd + C = 0$, where d is the slant range to the ground point and

$$A = \left(\frac{F_x}{a}\right)^2 + \left(\frac{F_y}{a}\right)^2 + \left(\frac{F_z}{b}\right)^2$$

$$B = \frac{F_x}{a^2} R\cos(\phi)\cos(\lambda) + \frac{F_y}{ab} R\sin(\phi)\sin(\lambda) + \frac{F_z}{b^2} R\sin(\phi)$$

$$C = \left(\frac{R\cos(\phi)\cos(\lambda)}{a}\right)^2 + \left(\frac{R\sin(\phi)\sin(\lambda)}{b}\right)^2 + \left(\frac{R\sin(\phi)}{b}\right)^2 - 1$$

where a and b are the Earth's semi-major and semi-minor axes (equatorial and polar radii), respectively.

9. Compute the two solutions to the quadratic:

$$D_1 = \frac{-B + \sqrt{B^2 - AC}}{A}$$

$$D_2 = \frac{-B - \sqrt{B^2 - AC}}{A}$$

and set $d = \mathrm{MIN}(D_1, D_2)$.

10. Compute the look point right ascension (λ_p):

$$U_x = R\cos(\phi)\cos(\lambda) + F_x d$$

$$U_y = R\cos(\phi)\sin(\lambda) + F_y d$$

$$\lambda_p = \tan^{-1}\left(\frac{U_y}{U_x}\right)$$

11. Compute the geodetic latitude of the look point (ϕ_{GD}):

$$U_z = R\sin(\phi) + F_z d$$

$$U_{xy} = U_x\cos(\lambda_p) + U_y\sin(\lambda_p)$$

$$\phi_{GD} = \tan^{-1}\left(\frac{U_z a^2}{U_{xy} b^2}\right)$$

12. Compute the GHA (right ascension of Greenwich) at the specified time using subroutine JGRENHA (RSL routine) or JGHAX. This computation requires use of the

time coefficients file. The current time must first be converted to A.1 Julian date by calling RSL subroutine CALAMJD.

13. Compute the longitude of the view point from right ascension and GHA:

$$lon = \lambda - GHA$$

Verify that *lon* is between -180 and +180 degrees.

E.4.2 Locating the Scene Center and Scene Corners

The scene center is forced to be at a nominal WRS latitude (one of 125 different latitudes) by interpolating to find the time of crossover. Once the scene center is found, the trailing edge corners are found by backing up in time TBD seconds (assuming the nominal scan time for one scene is 23.92 seconds, this would be half that or 11.96 seconds) and interpolating on the trailing edge corner points to find the locations at that time (the scene start time). Similarly, the leading edge corners are found by moving forward to TBD seconds after the scene center time.

1. Check the computed geodetic latitude of the scan center (optical axis) at the current time against a table of nominal WRS scene center geodetic latitudes (there are 125 total latitudes in the WRS) to determine if a crossover has occurred between the current time and the previous time. Table E–3 lists the WRS latitudes. Each latitude is associated with two possible row numbers, except for rows 246 and 122, which are the northern-most and southern-most latitudes, respectively. If crossover has occurred, interpolate to find the time when the geodetic latitude of the scan center look point matches the nominal scene center latitude. Use an N-point Lagrange interpolator (where N is at least 4). The latitude of the look point will be the nominal scene center latitude. Compute the longitude of the look point as specified in the previous algorithm. The computed longitude will usually be offset from the nominal scene center longitude.

2. Using the computed scene center latitude and longitude, determine the closest WRS path and row numbers through table look-up (or compute using the path/row algorithm). The row number will be one of two possible values, depending on whether this is an ascending or descending pass. Path number will depend on the longitude.

3. Find the trailing edge corners of the scene by backing up TBD seconds prior to the scene center time. Interpolate to find the spacecraft position and attitude at that time. Use the trailing edge along-scan and cross-scan angles (see tabulation in Section E.4.1) to compute the look point geodetic latitude and longitude values for the corner points.

4. Compute the time of the leading edge corners by going forward in time TBD seconds and interpolating to get position and attitude at that time. Use the leading edge along-scan and cross-scan angles to compute the corner look points.

Table E–3. WRS Latitudes (1 of 2)

WRS	Row	Latitude (deg)	Latitude (deg/min)	WRS	Row	Latitude (deg)	Latitude (deg/min)
246	–	81.8544	81/51	61	183	-1.4465	-1/27
245	247	81.7286	81/44	62	182	-2.8929	-2/54
244	248	81.3621	81/22	63	181	-4.3393	-4/20
1	243	80.7836	80/47	64	180	-5.7855	-5/47
2	242	80.0300	80/02	65	179	-7.2316	-7/14
3	241	79.1375	79/08	66	178	-8.6774	-8/41
4	240	78.1373	78/08	67	177	-10.1230	-10/07
5	239	77.0545	77/03	68	176	-11.5682	-11/34
6	238	75.9079	75/54	69	175	-13.0132	-13/01
7	237	74.7119	74/43	70	174	-14.4577	-14/27
8	236	73.4770	73/29	71	173	-15.9017	-15/54
9	235	72.2114	72/13	72	172	-17.3453	-17/21
10	234	70.9211	70/55	73	171	-18.7884	-18/47
11	233	69.6107	69/37	74	170	-20.2309	-20/14
12	232	68.2837	68/17	75	169	-21.6727	-21/40
13	231	66.9430	66/57	76	168	-23.1139	-23/07
14	230	65.5907	65/35	77	167	-24.5543	-24/33
15	229	64.2287	64/14	78	166	-25.9940	-25/60
16	228	62.8582	62/51	79	165	-27.4328	-27/26
17	227	61.4806	61/29	80	164	-28.8708	-28/52
18	226	60.0966	60/06	81	163	-30.3077	-30/18
19	225	58.7071	58/42	82	162	-31.7437	-31/45
20	224	57.3126	57/19	83	161	-33.1787	-33/11
21	223	55.9139	55/55	84	160	-34.6125	-34/37
22	222	54.5112	54/31	85	159	-36.0450	-36/03
23	221	53.1051	53/06	86	158	-37.4763	-37/29
24	220	51.6958	51/42	87	157	-38.9063	-38/54
25	219	50.2836	50/17	88	156	-40.3347	-40/20
26	218	48.8687	48/52	89	155	-41.7617	-41/46
27	217	47.4514	47/27	90	154	-43.1869	-43/11
28	216	46.0319	46/02	91	153	-44.6104	-44/37
29	215	44.6104	44/37	92	152	-46.0319	-46/02
30	214	43.1869	43/11	93	151	-47.4514	-47/27

Table E–4. WRS Latitudes (2 of 2)

WRS	Row	Latitude (deg)	Latitude (deg/min)	WRS	Row	Latitude (deg)	Latitude (deg/min)
31	213	41.7617	41/46	94	150	-48.8687	-48/52
32	212	40.3347	40/20	95	149	-50.2836	-50/17
33	211	38.9063	38/54	96	148	-51.6958	-51/42
34	210	37.4763	37/29	97	147	-53.1051	-53/06
35	209	36.0450	36/03	98	146	-54.5112	-54/31
36	208	34.6125	34/37	99	145	-55.9139	-55/55
37	207	33.1787	33/11	100	144	-57.3126	-57/19
38	206	31.7437	31/45	101	143	-58.7071	-58/42
39	205	30.3077	30/18	102	142	-60.0966	-60/06
40	204	28.8708	28/52	103	141	-61.4806	-61/29
41	203	27.4328	27/26	104	140	-62.8582	-62/51
42	202	25.9940	25/60	105	139	-64.2287	-64/14
43	201	24.5543	24/33	106	138	-65.5907	-65/35
44	200	23.1139	23/07	107	137	-66.9430	-66/57
45	199	21.6727	21/40	108	136	-68.2837	-68/17
46	198	20.2309	20/14	109	135	-69.6107	-69/37
47	197	18.7884	18/47	110	134	-70.9211	-70/55
48	196	17.3453	17/21	111	133	-72.2114	-72/13
49	195	15.9017	15/54	112	132	-73.4770	-73/29
50	194	14.4577	14/27	113	131	-74.7119	-74/43
51	193	13.0132	13/01	114	130	-75.9079	-75/54
52	192	11.5682	11/34	115	129	-77.0545	-77/03
53	191	10.1230	10/07	116	128	-78.1373	-78/08
54	190	8.6774	8/41	117	127	-79.1375	-79/08
55	189	7.2316	7/14	118	126	-80.0300	-80/02
56	188	5.7855	5/47	119	125	-80.7836	-80/47
57	187	4.3393	4/20	120	124	-81.3621	-81/22
58	186	2.8929	2/54	121	123	-81.7286	-81/44
59	185	1.4465	1/27	122	–	-81.8544	-81/51
60	184	0	0/0				

Acronyms

ABS	as-built specification
ACCA	automatic cloud cover assessment
AOS	acquisition of signal
API	application programming interface
ASCII	American Standard Code for Information Interchange
BCH	Bose-Chaudhuri-Hocquenghem
BER	bit error rate
CADU	channel access data unit
CASE	computer-aided software engineering
CCSDS	Consultative Committee for Space Data Systems
CDE	Cooperative Development Environment
CNMOS	Consolidated Network and Mission Operations Support
COTS	commercial off-the-shelf
CPU	central processing unit
CRC	cyclic redundancy check
CRUD	create, retrieve, update, and delete
CSC	Computer Sciences Corporation
DAA	data availability acknowledgment
DAAC	Distributed Active Archive Center
DAN	data availability notice
DAT	digital audio tape
DBAR	database access routine
DBMS	database management system
DDA	data delivery acknowledgment
DDF	Data Distribution Facility
DDN	data delivery notice
DFCB	data format control book
DLT™	Digital Linear Tape

DSI	delivered source instruction
ECS	EOS Core System
EDC	EROS Data Center
EOC	end of contact
EOL	end of line
ER	entity relationship
ERD	entity relationship diagram
EROS	Earth Resources Observation System
ETM+	Enhanced Thematic Mapper Plus
FDDI	fiber-distributed data interface
FIFO	first-in-first-out
FILO	first-in-last-out
F&PS	functional and performance specification
FTP	File Transfer Protocol
GCI	geocentric inertial
GHA	Greenwich hour angle
GUI	graphical user interface
HDF	hierarchical data format
HDS	horizontal display shift
HP	Hewlett-Packard
HPDI	high-speed peripheral device interface
HWCI	hardware configuration item
IAS	Image Assessment System
IDD	interface definition document
IDPS	image data processing subsystem
I/O	input/output
IP	Internet Protocol
IPC	interprocess communication
LDTS	LPS data transfer subsystem

LGS	Landsat Ground Station
LMAS	Lockheed Martin Astro Space
LPS	Landsat 7 Processing System
MACS	management and control subsystem
MFPS	major frame processing subsystem
MMAS	Martin Marietta Astro Space
MMO	Mission Management Office
MOC	Mission Operations Center
MO&DSD	Mission Operations and Data Systems Directorate
MSCD	mirror scan correction data
MTBF	mean time between failure
MTTRes	mean time to restore
MWD	moving window display
NASA	National Aeronautics and Space Administration
NCSA	National Center for Supercomputing Applications
NOAA	National Oceanic and Atmospheric Administration
PCD	payload correction data
PCDS	PCD processing subsystem
PRN	pseudorandom noise
Q&A	quality and accounting
RAID	redundant array of independent disks
RAM	random access memory
RDCS	raw data capture subsystem
RDPS	raw data processing subsystem
RMA	reliability, maintainability, availability
RS	Reed-Solomon
RSL	Reusable Software Library
SCLF	search, check, lock, and flywheel
SDS	system design specification

SGI	Silicon Graphics, Inc.
SQL	Structured Query Language
SRS	software requirements specification
SSDM	SEAS System Development Methodology
SWCI	software configuration item
TCP	Transmission Control Protocol
USGS	United States Geological Survey
UTC	universal time coordinated
VCDU	virtual channel data unit
VCID	virtual channel identifier
VME	Versa-Module European
WRS	Worldwide Reference System